

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий  
Кафедра систем искусственного интеллекта

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ Г. М. Цибульский

«\_\_\_\_\_» \_\_\_\_\_ 20 \_\_\_\_ г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.02 «Информационные системы и технологии»

Проектирование и разработка мобильного приложения CityQuest на базе  
операционной системы Android

Руководитель \_\_\_\_\_  
подпись, дата

к. т. н. доцент

К. В. Раевич

Выпускник \_\_\_\_\_  
подпись, дата

А. А. Мазун

Выпускник \_\_\_\_\_  
подпись, дата

к. т. н. доцент

Д. А. Перфильев

Красноярск 2017

Продолжение титульного листа бакалаврской работы по теме  
«Проектирование и разработка мобильного приложения CityQuest на базе  
операционной системы Android»

Нормоконтролер \_\_\_\_\_  
подпись

М. А. Аникьева

## РЕФЕРАТ

ВКР по теме «Проектирование и разработка приложения CityQuest на базе операционной системе Android» содержит 45 страниц текстового документа, 40 рисунков, 15 используемых источников.

ANODROID, ANDRPOID ПРИЛОЖЕНИЕ, АРХИТЕКТУРА VIPER, МОБИЛЬНОЕ ПРИЛОЖЕНИЕ.

Цель работы – проектирование и разработка приложения CityQuest на базе операционной системе Android.

Проведен анализ требований, предъявляемых к приложению, рассмотрены аналогичные мобильные приложения. Проведено проектирование приложения, построены UML-диаграммы. На основе проектируемого приложения разработан прототип программы, выполняющий все поставленные задачи.

Приложение проходит тестовую эксплуатацию.

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий  
Кафедра систем искусственного интеллекта

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ Г. М. Цибульский  
подпись  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

**ЗАДАНИЕ**  
**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**  
**В форме бакалаврской работы**

Студенту Мазуну Александру Александровичу

Группа КИ13-15Б, направление 09.03.02 «Информационные системы и технологии», профиль 09.03.02.05 «Информационные системы и технологии в административном управлении».

Тема выпускной квалификационной работы «Проектирование и разработка мобильного приложения CityQuest на базе операционной системы Android».

Утверждена приказом по университету № 2567/с от 1 марта 2017 г.

Руководитель ВКР Раевич К. В. старший преподаватель кафедры СИИ.

Перечень разделов ВКР:

- введение;
- глава 1 Теоретическая часть;
- вывод по главе 1;
- глава 2 Проектирование приложения;
- вывод по главе 2;
- глава 3 Разработка приложения;
- вывод по главе 3;
- заключение;
- список использованных источников;
- приложение А. Техническое задание;
- приложение Б. Графический материал.

Руководитель ВКР

\_\_\_\_\_  
подпись

К. В. Раевич

Задание принял к исполнению

\_\_\_\_\_  
подпись

А. А. Мазун

## График

Выполнение выпускной квалификационной работы студентом направления 09.03.02 «Информационные системы и технологии», профиля 09.03.02.05 «Информационные системы и технологии в административном управлении». График выполнения выпускной квалификационной работы приведен в таблице 1.

Таблица 1 – График выполнения этапов ВКР

Наименование этапа	Срок выполнения этапа	Результат выполнения этапа	Примечание руководителя
Ознакомление с целью и задачами работы	06.03 – 12.03.2017	Краткое эссе по теме ВКР	Выполнено
Сбор литературных источников	13.03 – 19.03.2017	Список источников информации	Выполнено
Анализ собранных литературных источников	20.03 – 26.03.2017	Реферат о проблемно-предметной области	Выполнено
Уточнение и обоснование актуальности цели и задач ВКР	27.03 – 02.04.2017	Окончательная формулировка цели и задач ВКР	Выполнено
Подготовка доклада и презентации по теме ВКР	03.04 – 09.04.2017	Доклад и презентация по ВКР	Выполнено
Решение первой задачи	10.04 – 14.04.2017	Доклад и презентация по первой задаче ВКР	Выполнено
Решение второй задачи	15.04 – 19.04.2017	Доклад и презентация по второй задаче ВКР	Выполнено
Решение третьей задачи	20.04 – 24.05.2017	Доклад и презентация по третьей задаче ВКР	Выполнено
Компоновка отчета по результатам решения задач ВКР	25.05 – 02.06.2017	Отчет по результатам решения ВКР	Выполнено
Нормконтроль	03.06 - 15.06.2017	Пояснительная записка, презентация к ВКР	Выполнено
Предварительная защита результатов ВКР	06.06.2017	Доклад и презентация о проделанной работе	Выполнено
Защита ВКР	21.06.2017	Доклад и презентация по результатам бакалаврской работы	

Руководитель ВКР

\_\_\_\_\_

подпись

К. В. Раевич

Студент гр. КИ13-156

\_\_\_\_\_

подпись

А. А. Мазун

## СОДЕРЖАНИЕ

Введение.....	4
Глава 1 Теоретическая часть.....	6
1.1 Структурный анализ предметной области .....	6
1.1.1 Структурный анализ системы. Модель AS-IS (как есть).....	7
1.1.2 Структурный анализ системы. Модель TO-BE (как будет).....	9
1.2 Обзор приложения «Ловушка».....	12
1.3 Обзор приложения «СибКвест».....	14
1.5 Вывод к главе 1.....	16
Глава 2 Проектирование приложения.....	18
2.1 Архитектура.....	18
2.2 UML-моделирование Use-case диаграмма .....	23
2.2.1 UML-моделирование диаграммы деятельности .....	24
2.2.2 UML-моделирование диаграммы классов.....	25
2.2.3 UML-моделирование диаграммы последовательности .....	27
2.2.4 UML-моделирование диаграммы компонентов.....	28
2.2.5 UML-моделирование диаграммы развертывания.....	30
2.3 Вывод к главе 2.....	31
Глава 3 Разработка приложения .....	32
3.1 Интерфейс программы.....	32
3.1.1 Регистрация в приложении .....	33
3.1.2 Авторизация в приложении .....	34
3.1.3 Восстановление пароля .....	35
3.1.4 Меню приложения .....	36

3.1.5 Карта квестов.....	37
3.1.6 Бронирование .....	39
3.1.7 Мои брони.....	42
3.1.8 Экран Техподдержки .....	42
Вывод к главе 3.....	43
Заключение .....	44
Список сокращений .....	45
Список использованных источников .....	46
Приложение А Техническое задание .....	48
Приложение Б Графический материал .....	51



## ВВЕДЕНИЕ

В настоящее время каждый человек владеет мобильным гаджетом, который содержит множество приложений, увеличивающие функционал устройства. Будь то файловый менеджер или мессенджер (Viber, WhatsUp). Без них уже невозможно представить себе современное устройство связи. Существует множество разработчиков приложений, а их продукцию легко могут установить пользователи через различные площадки (PlayMarket, AppStore).

Рынок этих электронных устройств постоянно растет, но самая популярная операционная система мобильных устройств является Android. Объемы продаж устройств Android и количество загрузок Android-приложений растут экспоненциально. Мобильные телефоны Android первого поколения появились на рынке в октябре 2008 года. Согласно результатам исследования рынка, а уже в июле 2011 года смартфоны Android занимали 41,8 % рынка смартфонов. Количество приложений, загружаемых с Google Play, исчисляется миллиардами. Ежедневно активируются более 500 000 устройств Android [1]. В настоящий момент эта цифра переваливает за 80 %. Поэтому разработка приложений для мобильных гаджетов является популярным решением для многих компаний, в основном их создают с целью автоматизации различных процессов и удобства клиентов, которые могут легко увидеть изменение в условиях предоставления товаров и услуг и выполнить онлайн заказ или бронирование.

Рынок услуг постоянно расширяется в большой степени заметно это по развитию предоставления развлечений. С недавних пор во многих крупных городах России появилось новое веяние под названием «квест в реальности». При прохождении квеста команда от двух до пяти человек на определенное время попадает в нестандартную ситуацию и пытается разрешить ее любым путем. В России первый «квест в реальности» возник в 2012 году в городе Екатеринбурге, а дальше это направление начало развиваться семимильными

шагами. В 2015 году в городе Красноярске насчитывалось около десятка площадок для проведения этих развлечений, на начало 2017 года их количество приближается к сотне. Огромное многообразие приводит к конкуренции, поэтому многие компании привлекают современные технологии для расширения потенциальных клиентов и усовершенствования сервиса по предоставлению услуг на проведение.

Целью выпускной квалификационной работы является проектирование и разработка мобильного приложения CityQuest на базе операционной системы Android.

Для достижение поставленной цели необходимо решить задачи:

- Структурный анализ информационных процессов, обзор существующих мобильных приложений.
- Проектирование мобильного приложения CityQuest.
- Разработка мобильного приложения CityQuest.

## **Глава 1 Теоретическая часть**

### **1.1 Структурный анализ предметной области**

Методология SADT представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель SADT отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями. Основные элементы этой методологии основываются на следующих концепциях:

- графическое представление блочного моделирования. Графика блоков и дуг SADT-диаграммы отображает функцию в виде блока, а интерфейсы входа/выхода представляются дугами, соответственно входящими в блок и выходящими из него;
- строгость и точность.

Выполнение правил SADT требует достаточной строгости и точности, не накладывая в то же время чрезмерных ограничений на действия аналитика. Правила SADT включают:

- ограничение количества блоков на каждом уровне декомпозиции (правило 3-6 блоков);
- связность диаграмм (номера блоков);
- уникальность меток и наименований (отсутствие повторяющихся имен);
- синтаксические правила для графики (блоков и дуг);
- разделение входов и управлений (правило определения роли данных);
- отделение организации от функции, т.е. исключение влияния организационной структуры на функциональную модель.

Методология SADT может использоваться для моделирования широкого круга систем и определения требований и функций, а затем для разработки системы, которая удовлетворяет этим требованиям и реализует эти функции.

Для уже существующих систем SADT может быть использована для анализа функций, выполняемых системой, а также для указания механизмов, посредством которых они осуществляются [1].

Представление процесса составления заявки представлен в виде SADT-Модели. SADT-модель – модель, которая дает полное, точное и адекватное описание системы, имеющее конкретное назначение.

Структурный анализ систем подразумевает создание двух видов функциональных моделей предметной области:

- модели существующей организации процесса AS-IS (как есть);
- модели новой организации процесса TO-BE (как будет).

### **1.1.1 Структурный анализ системы. Модель AS-IS (как есть)**

Модель AS-IS позволяет определить неэффективные места существующего на момент моделирования процесса, оценить, насколько глубоким изменениям необходимо подвергнуть существующую структуру организации системы.

Такую модель часто называют функциональной и выполняют с использованием различных графических нотаций и CASE-средств. На этапе построения модели важным считается строить максимально приближенную к действительности модель, основанную на реальных потоках процессов, а не на их идеализированном представлении.

Далее на рисунках 1.1 и 1.2 представлена модель AS-IS процесса создания бронирования.

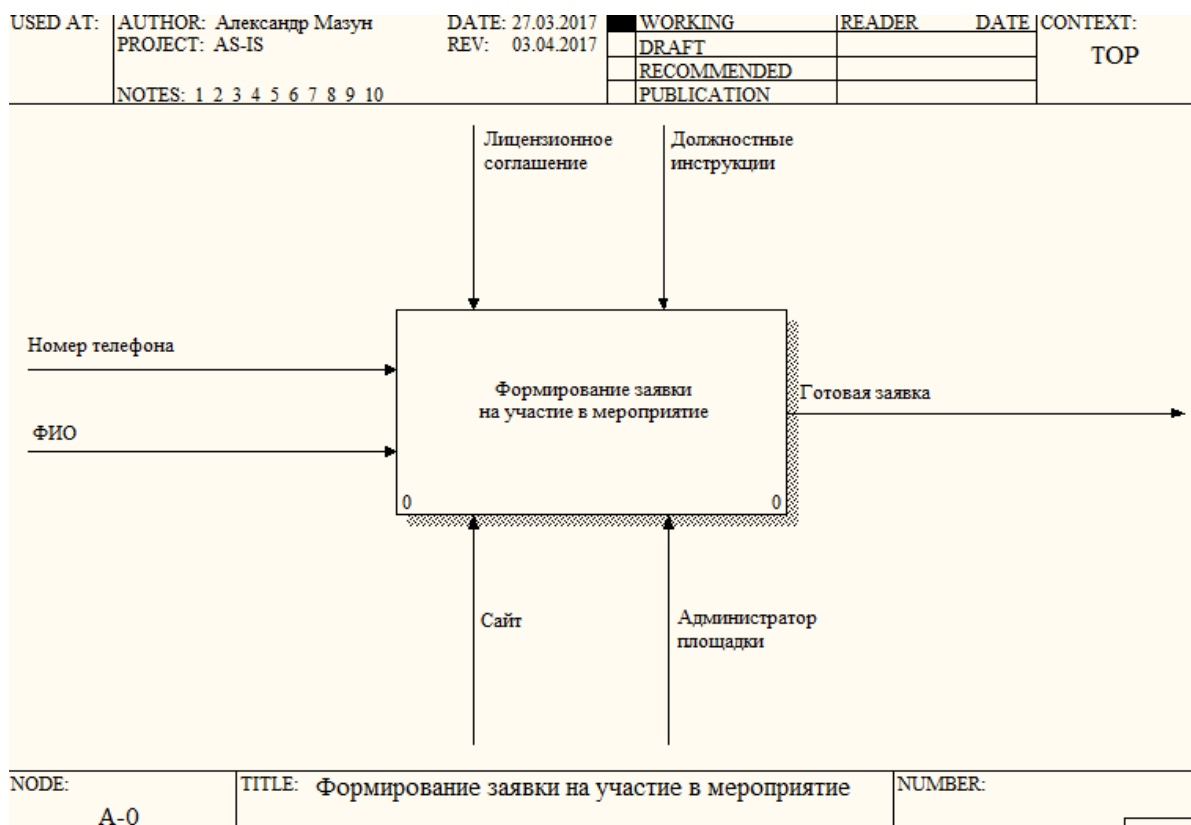


Рисунок 1.1 — Композиционная диаграмма «Формирование заявки на участие в мероприятии»

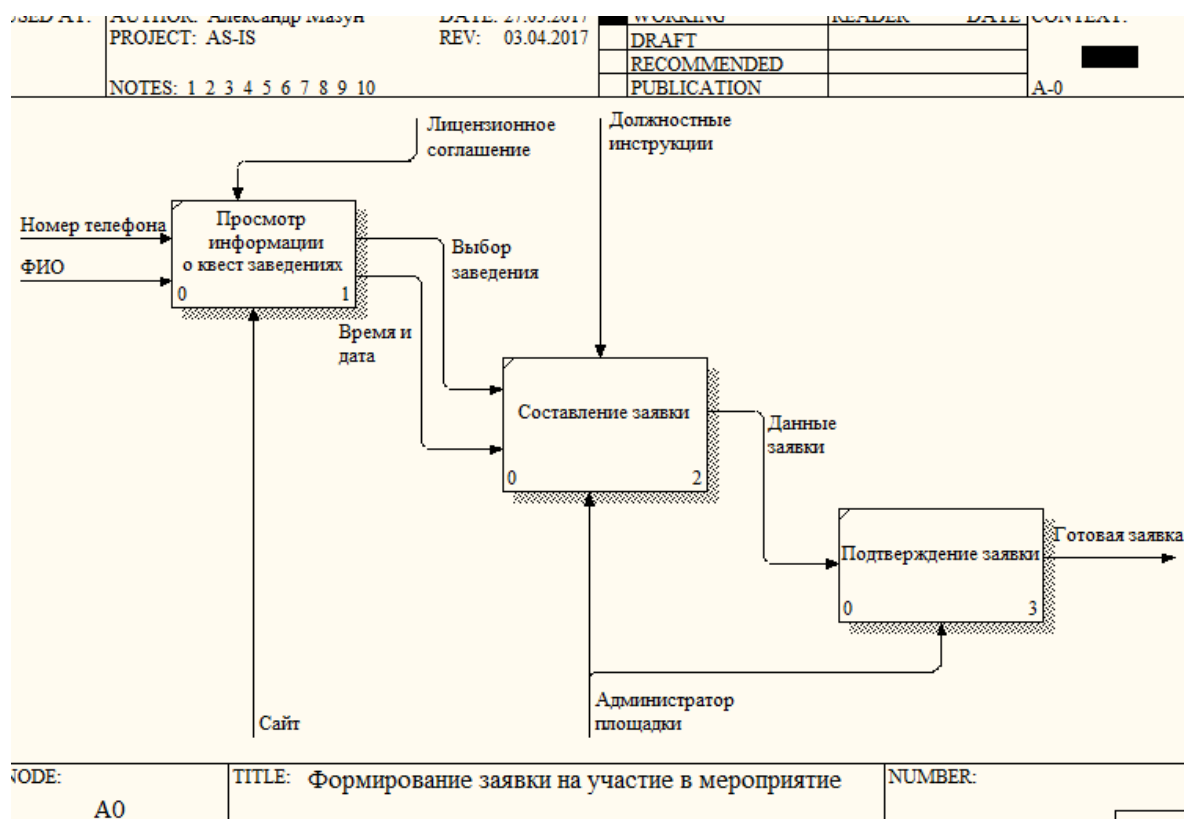


Рисунок 1.2 — Декомпозиция блока «Формирование заявки на участие в мероприятии»

Процесс оформление заявки на участие занимает определенное время клиента, а также рабочее время администратора мероприятия.

Аспекты, которые можно улучшить путем внедрения приложения:

- Наличие человеческого фактора (неверное указание данных о клиенте: ФИО, возраст, мобильный телефон).
- Упразднение обязанностей администратора по бронирование (оформление заявки).

### 1.1.2 Структурный анализ системы. Модель ТО-ВЕ (как будет)

Анализ в модели «AS-IS» показал, какие процессы можно улучшить. Построение модели «ТО-ВЕ» показана на рисунках 1.3–1.7.

На данных диаграммах показан процесс после ввода приложение в эксплуатацию.

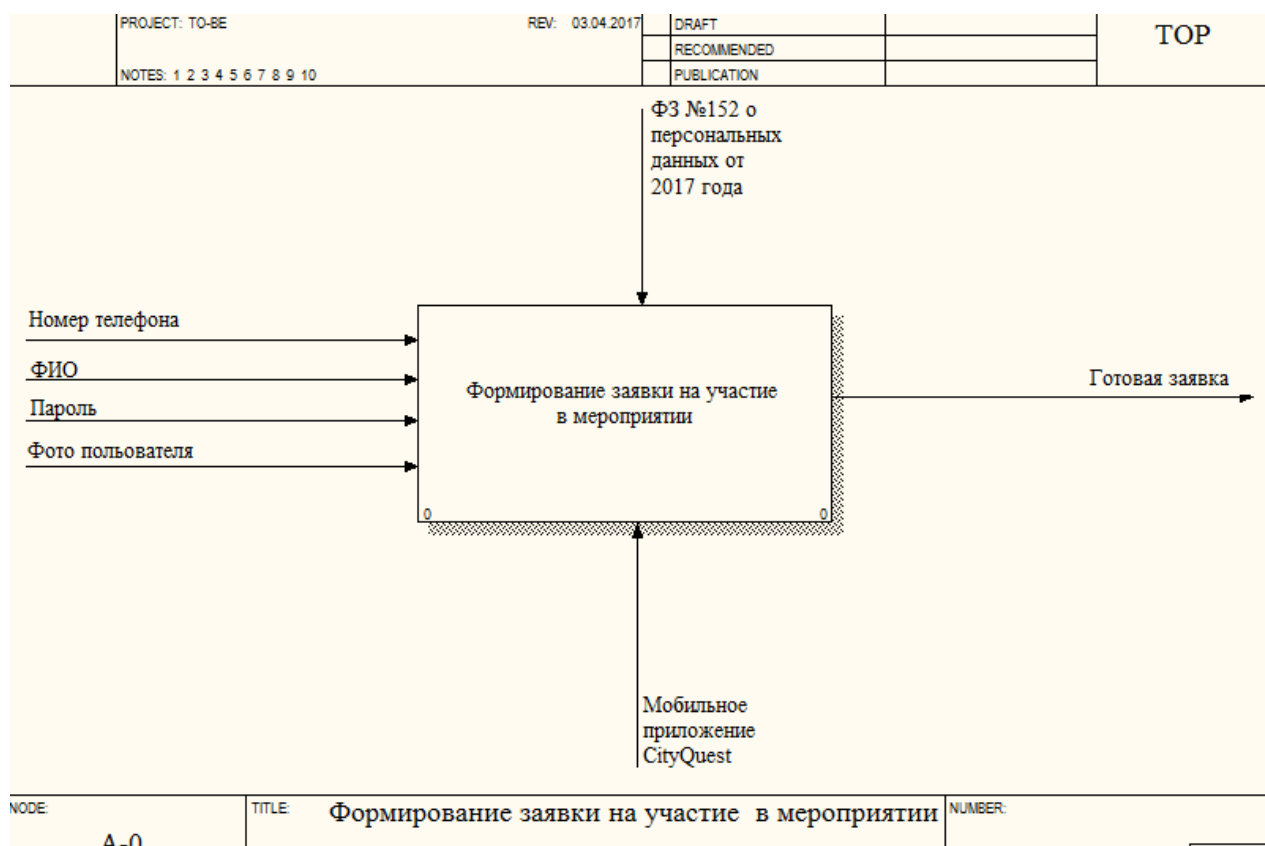


Рисунок 1.3 — Композиционная диаграмма «Формирование заявки на участие в мероприятии»

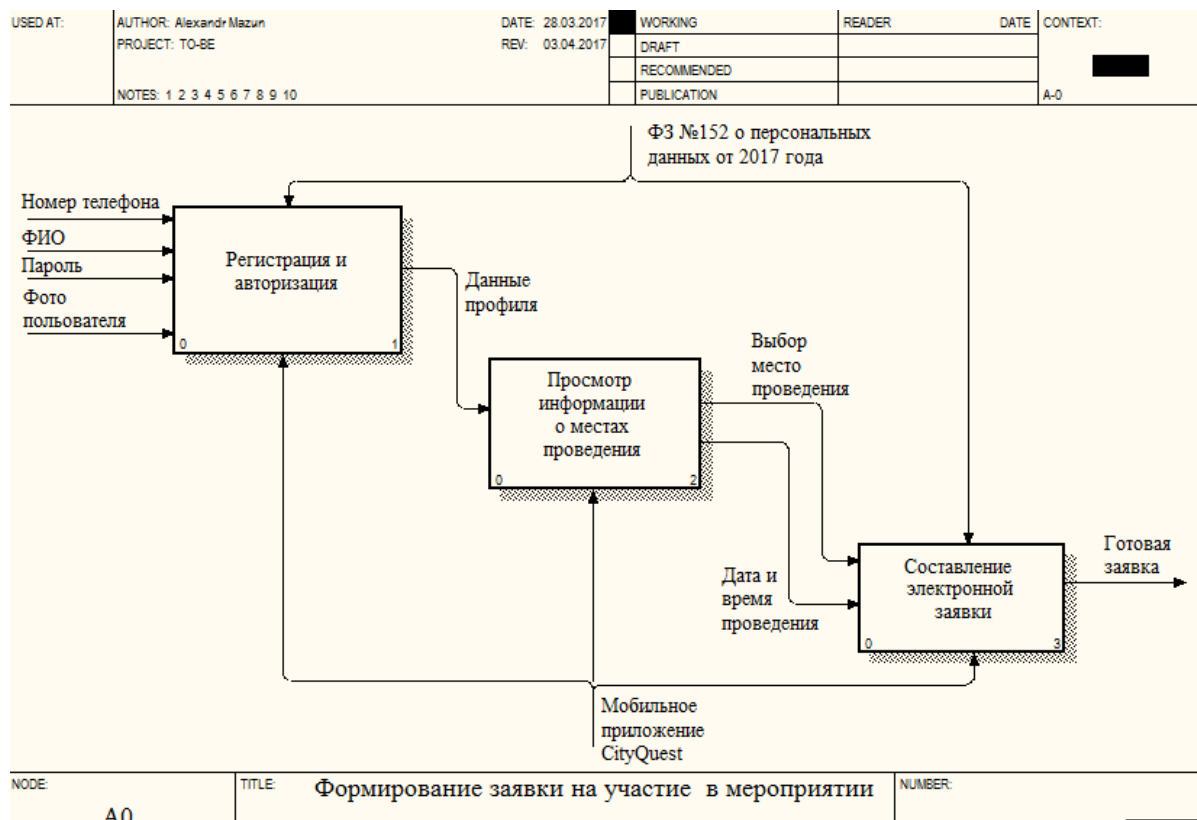


Рисунок 1.4 — Декомпозиция блока «Формирование заявки на участие в мероприятии»

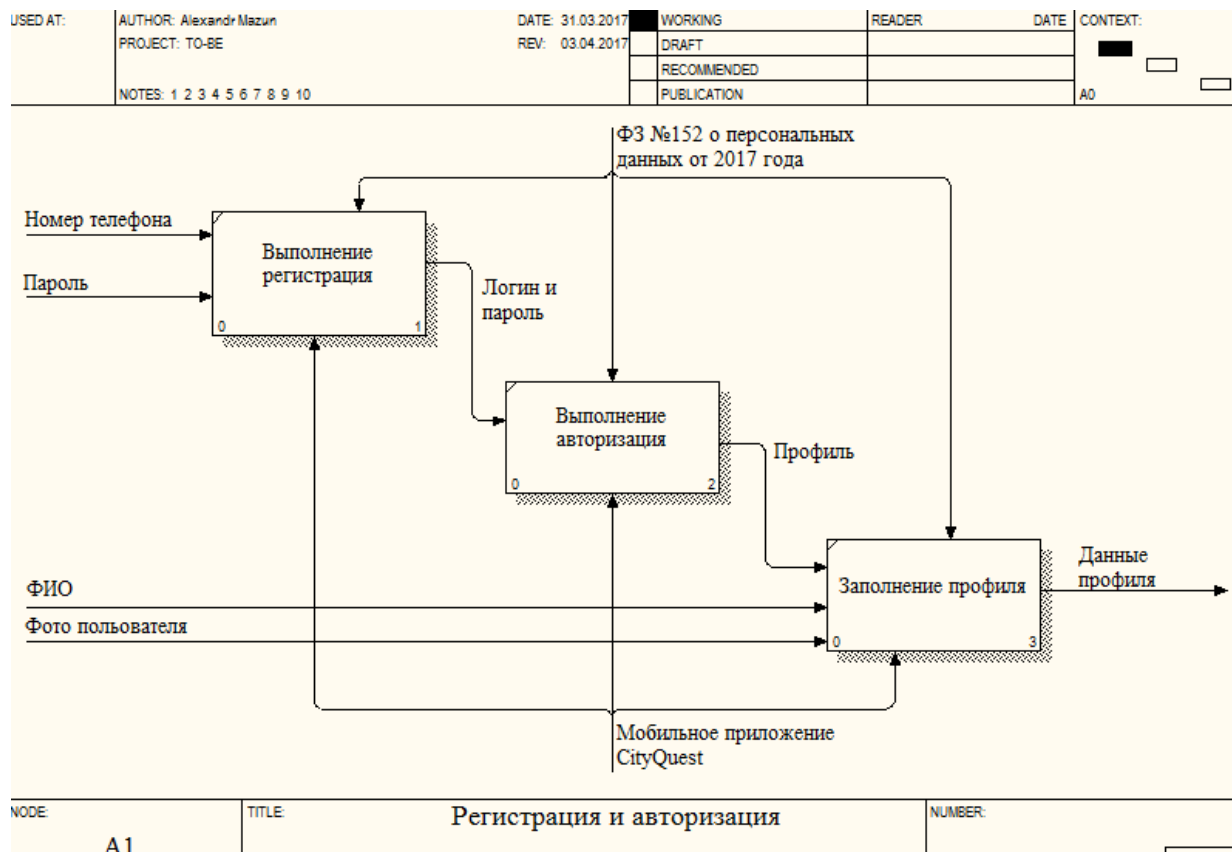


Рисунок 1.5 — Декомпозиция блока «Регистрация и авторизация»

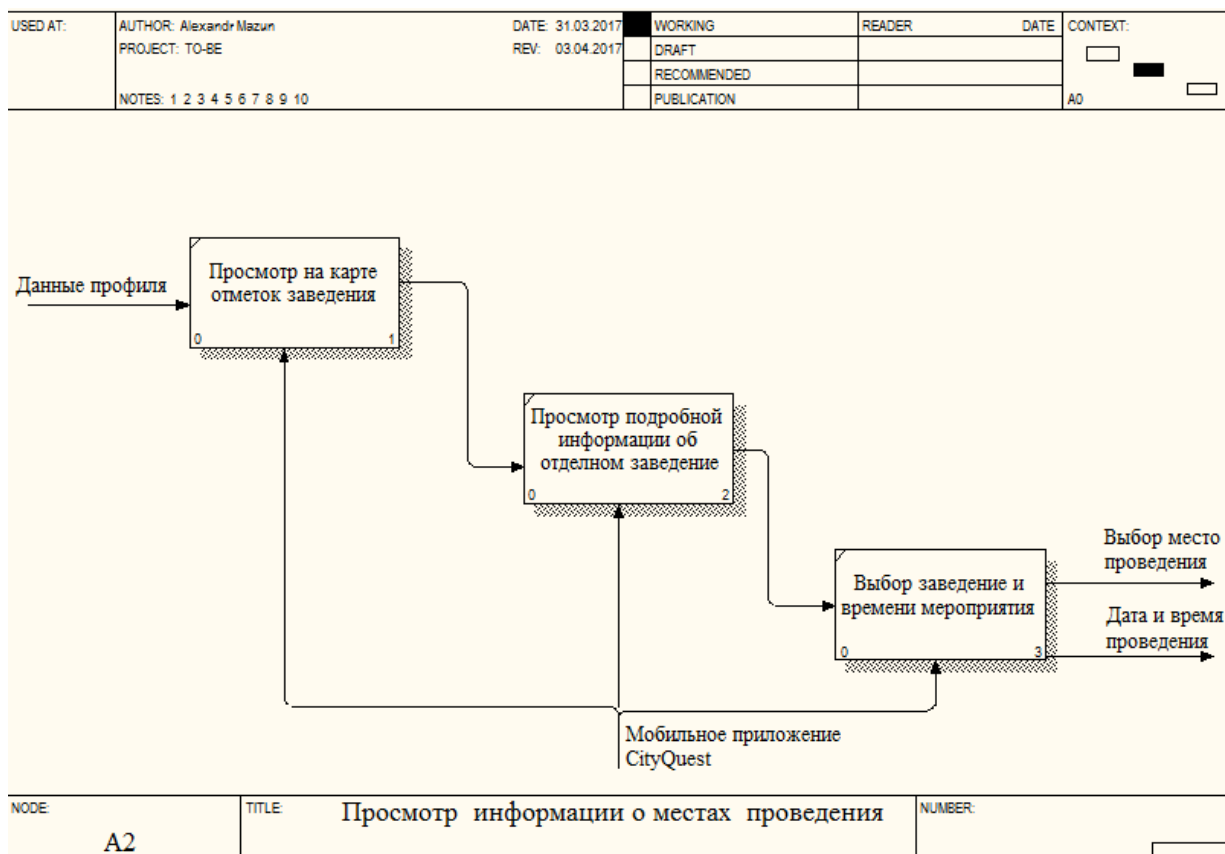


Рисунок 1.6 — Декомпозиция блока «Просмотр информации о местах проведения»

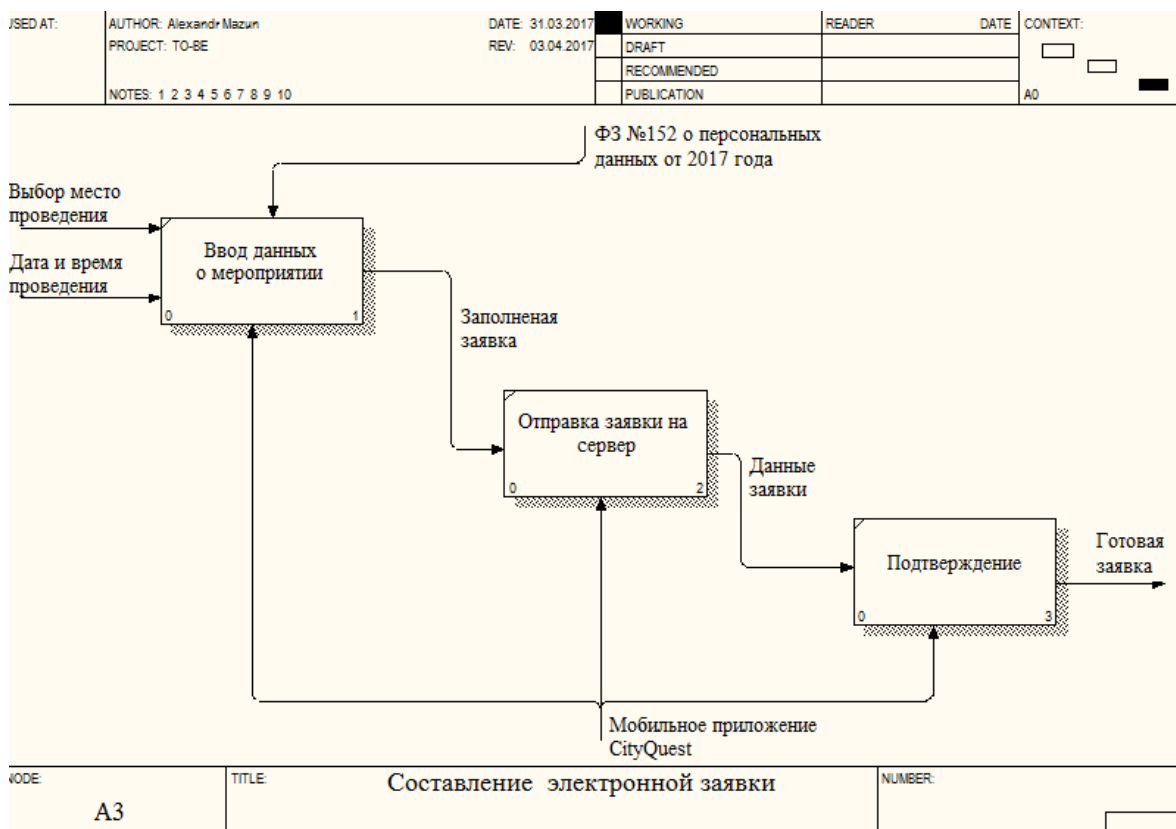


Рисунок 1.7 — Декомпозиция блока «Составление электронной заявки»



Результатом проведения анализа предметной области стала разработка структурной модели системы «AS-IS». Эта модель отражает, как происходит процесс бронирования «как есть».

Для оптимизации существующего бизнес-процесса разработана модель системы «TO-BE». Эта модель позволяет убрать человека от многих функций.

## 1.2 Обзор приложения «Ловушка»

Данное мобильное приложение разработано российской компанией «Artthemes» на базе операционной системы Android и распространено на площадке PlayMarket. Это мобильное приложение создано для поиска ближайших заведений в зависимости от местоположения пользователя и просмотра подробной информации о них. Приложение создано для обслуживания заведений одной компании в разных городах.



Рисунок 1.8 — Меню приложения



Рисунок 1.9 — Отображение информации о заведении

Немое кино Чарли Чаплина

ЛОВУШКА

БРОНИРОВАНИЕ КВЕСТА  
В РЯЗАНИ

Ваше имя\*

Ваш телефон\*

Выберите квест ▼

Укажите дату игры:\*

Укажите желаемое время игры:\*

Email - Куда выслать билет?\*

Промо - Код (Если есть)

Забронировать квест

Рисунок 1.10 — Экран бронирования

Функции приложения:

- Функция просмотра информации о заведении.
- Функция бронирования.
- Функция построение маршрута до заведения.

Минусы приложения:

- Не имеет функции отображения заведений на карте.
- Не имеет функции онлайн оплаты.

### 1.3 Обзор приложения «СибКвест»

Данное мобильное приложение разработано российской компанией «Traveler Today» на базе операционной системы Android и распространено на площадке PlayMarket.

Приложение представляет собой помощь в выборе квест комнат, путем отображение подробной информации о заведении и отображение его на карте.

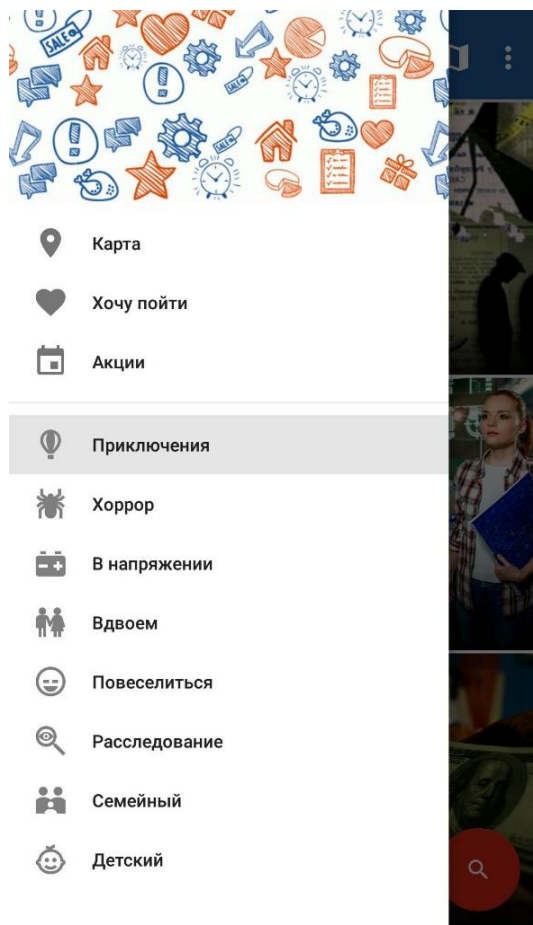


Рисунок 1.11 — Окно меню

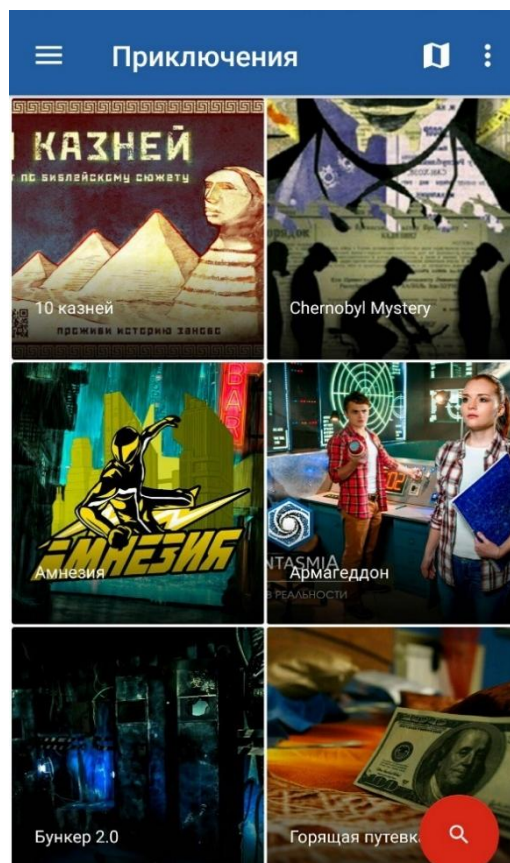


Рисунок 1.12 — Экран выбора квестов

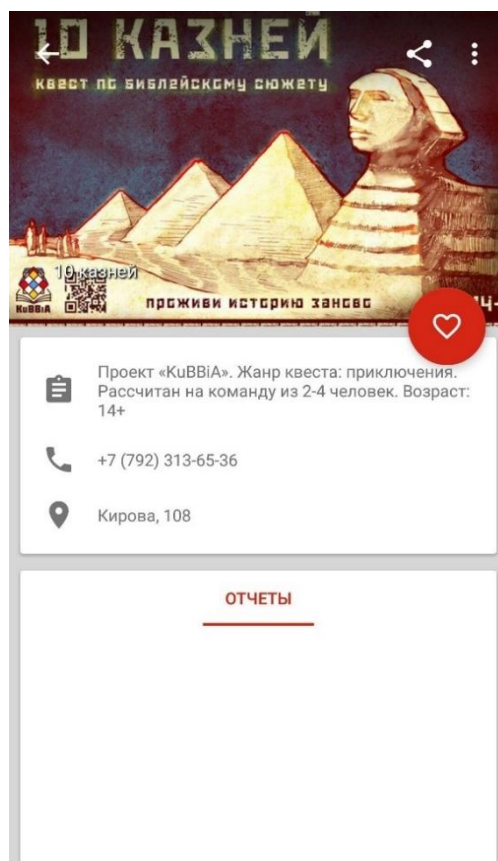


Рисунок 1.13 — Экран информации о квесте

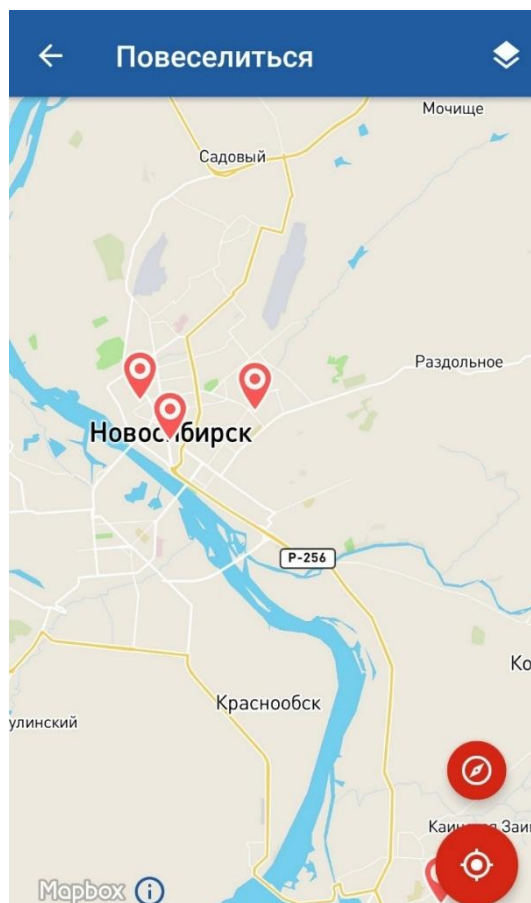


Рисунок 1.14 — Окно отображения заведений на карте

Функции приложения:

- Функция просмотра данных о заведении
- Функция написания и просмотра отзывов.
- Функция предоставления фотоотчетов.
- Функция просмотра заведений на карте (рисунок 1.10).

Минусы приложения:

- Не имеет функции онлайн бронирования.
- Не имеет функции онлайн оплаты.

## 1.5 Вывод к главе 1

В результате структурного анализа системы были выявлены основные функциональные требования:

- Регистрация пользователей.
- Отображение информации на карте.

- Отображение подробной информации о квестах.
- Бронирование.
- Оплата.

Обзор аналогов мобильных приложениях, таких как «СибКвест» и «Ловушка» позволил сформировать функции для разработки мобильного приложения. Функции онлайн-бронирования и отображение ближайших мест на карте являются удобными конечному пользователю и представляют обязательную реализацию в разрабатываемом программном обеспечении.

Для реализации всех требуемых функций необходимо спроектировать и разработать мобильное приложение на базе операционной системы Android.

## Глава 2 Проектирование приложения

### 2.1 Архитектура

В процессе создания проекта пишется огромное количество строк кода, если все описывается в одном паттерне с кодом становится сложно работать, для внесения изменений приходится постоянно искать где начинается та или иная функция, происходит постоянное повторение кода. Для упрощения работы создаются различные архитектуры приложений.

Существует различные архитектурные решения, в настоящий момент самые используемыми считается модель MVP и VIPER, выбранная для разрабатываемого приложения.

VIPER (View, Interactor, Presenter, Entity, Router) архитектура изначально создавалась для построения приложений на операционной системе IOS, но с течением времени разработчики стали использовать её для создания приложений на Android-устройства.

VIPER - это подход к архитектуре мобильных приложений, основанный на идеях Роберта Мартина, изложенных им в статье «The Clean Architecture» [2].

Основные достоинства и недостатки VIPER.

Достоинства:

- Повышение тестируемости Presentation-слоя приложений.
- Полная независимость модулей друг от друга - это позволяет независимо их разрабатывать и переиспользовать.
- Передача проекта другим разработчикам, либо внедрение нового, дается намного проще, так как общие подходы к архитектуре заранее определены.

Недостатки:

- Резкое увеличение количества классов в проекте, сложности при создании нового модуля.



- Отсутствие в открытом доступе набора конкретных рекомендаций, best practices и примеров сложных приложений [3].

Архитектура VIPER отображена на рисунке 2.1. Приложение делится на 3 слоя и 5 модулей, существует строгое разграничение между слоями программы.

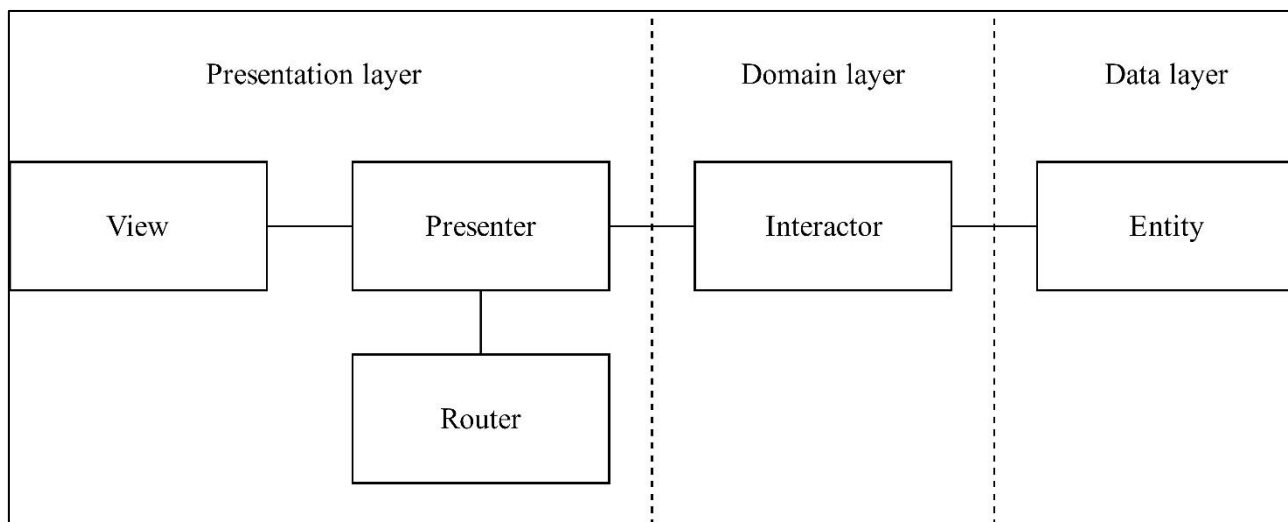


Рисунок 2.1 — Архитектура VIPER

Для более детального представления архитектуры VIPER, рассмотрим на модуле авторизации в приложении. Интерфейс авторизации пользователя, соответствующий View компоненту представлен на рисунке 2.2.

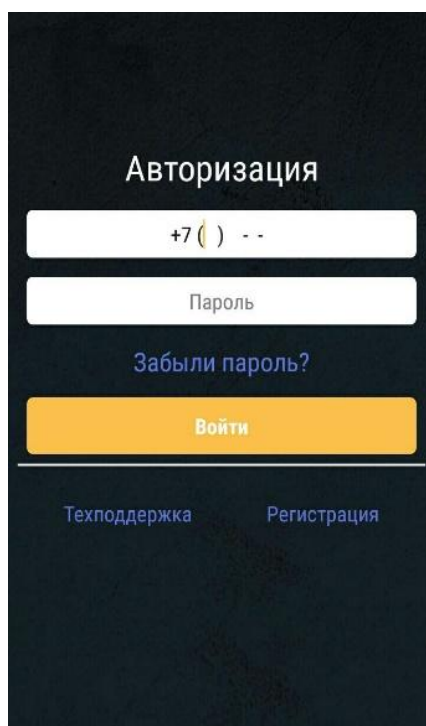


Рисунок 2.2 — Интерфейс авторизации



Для удобства восприятия связей между различными уровнями приложения, используем диаграмму классов (рисунок 2.3). Диаграмма классов одна из диаграмм UML, отображает классы системы, их атрибуты, методы и взаимосвязи между ними.

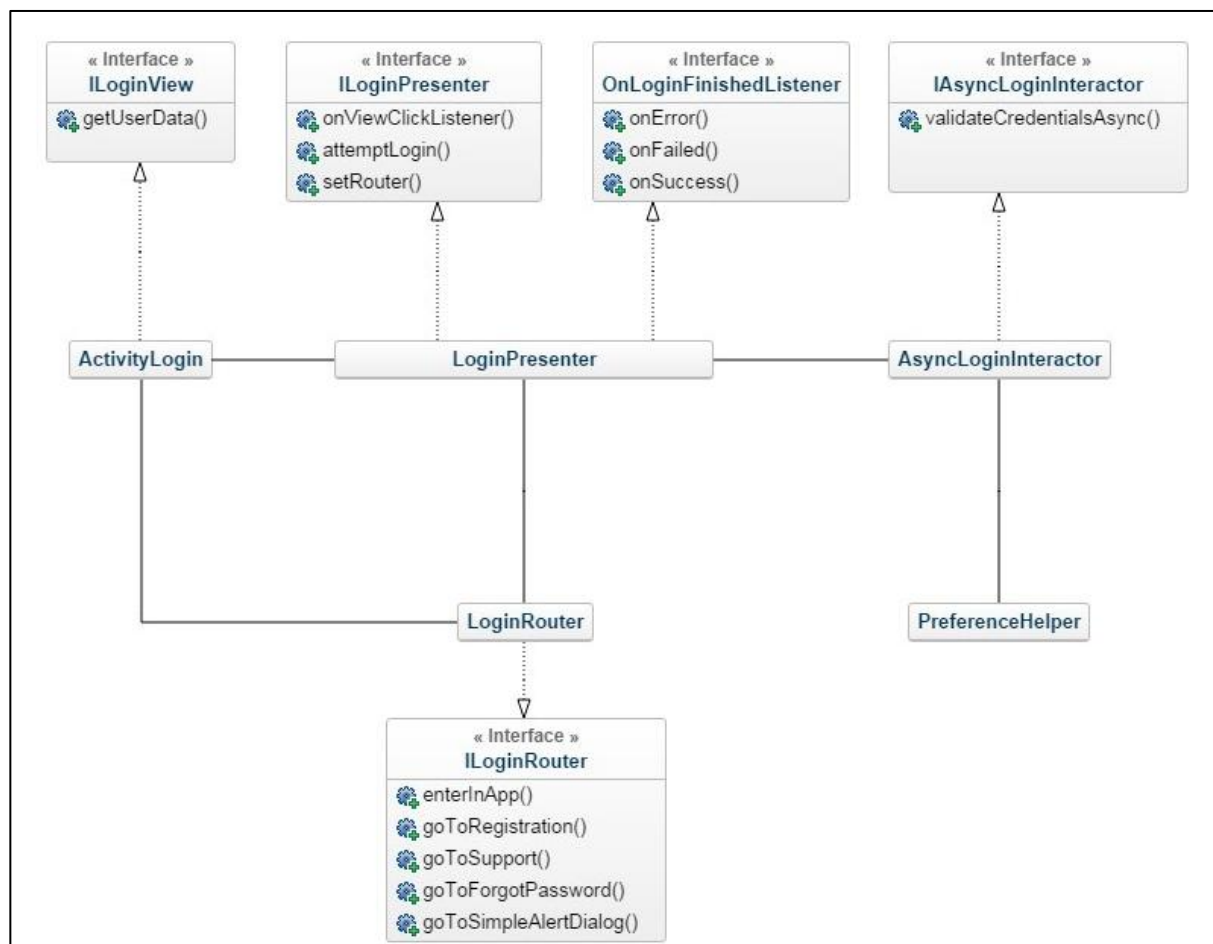


Рисунок 2.3 — Диаграмма классов

В диаграмме отображены различные классы, выполняющие следующие функции:

- Класс ActivityLogin(View) – содержит интерфейс ввода данных.
- Класс LoginPresenter(Presenter) – определяет куда отдать управление и передать данные.
- Класс LoginRouter(Router) – отвечает за переключение между различными экранами.
- Класс AsyncLoginInteractor(Interactor) – содержит в себе бизнес-логику приложения.

- Класс `PerferenceHelper(Entity)` – репозиторий приложения.

## View

Представляет собой интерфейс, с которым взаимодействует пользователь на экране своего мобильного устройства. Содержит то, что передал ему `Presenter` и ретранслирует в `Presenter` данные введение в нем. На рисунке 2.4 показан интерфейс с функцией передачи данных от `View` к `Presenter`.

```
public interface ILoginView {  
    void getUserData();  
}
```

Рисунок 2.4 — Представление View

## Presenter

Этот модуль содержит логику управления, является неким “проводником”, между представлением и бизнес-логикой (который решает куда передать управление). Он определяет запрос, поступающий со `View`, и решает, куда отправить его дальше в `Router` для изменения окна `Activity` или же передает функцию в `Interactor`. Представлен в виде оператора `switch-case`. На рисунке 2.5 представлены основные функции `Presenter` такие как: определение клика по `View`, передача данных в `Interactor`, инициализация `Router`.

```
public interface ILoginPresenter {  
    void onViewClickListener(int id);  
    void attemptLogin(String username, String password);  
    void setRouter(LoginRouter loginRouter);  
}
```

Рисунок 2.5 — Представление Presenter

## Router

Осуществляет переключения между экранами приложения. На рисунке 2.6 представлен интерфейс `Router`, он выполняет второстепенную функцию и является лишь инструментом.

```
public interface ILoginRouter {
    void enterInApp();
    void goToRegistration();
    void goToSupport();
    void goToForgotPassword();
    void goToSimpleAlertDialog(String msg);
}
```

Рисунок 2.6 — Представление Router

## Interactor

Включает в себя бизнес-логику для управления объектами данных (Entity). Функции, выполняющиеся в Interactor, не зависят от пользовательского интерфейса. На рисунке 2.7 представлен интерфейс с функцией запроса на сервер.

```
public interface IAsyncLoginInteractor {
    void validateCredentialsAsync(Context context, OnLoginFinishedListener listener, String username, String password);
}
```

Рисунок 2.7 – Представление Interactor

## Entity

Это представление объектов данных, которыми может управлять только Interacor. Он никогда не передает данные уровню представления. На диаграмме классов Entity представлена как PreferenceHelper.

Безусловно, построение приложения на архитектуре VIPER требует больше времени и большего количества задействованных классов, но приложение, построенное таким образом, представляется более простым и понятным с точки зрения чтения кода. Вся логика запросов на сервер и действий самого приложения отделена от пользовательского интерфейса и не содержится в одном файле (классе), это позволяет легче разобраться в коде и внедрить новый функционал добавляя элементы и новые классы, не переписывая весь код одного экрана.

## 2.2 UML-моделирование Use-case диаграмма

Визуальное моделирование в UML(Unified Modeling Language) можно представить, как некоторый процесс поуровневого спуска от наиболее общей и абстрактной концептуальной модели исходной системы к логической, а затем и к физической модели соответствующей программной системы.

Каждый сценарий использования сосредотачивается на описании того, как достигнуть цели или задачи. Для большинства программных проектов это означает, что потребуется множество сценариев использования, чтобы определить необходимый набор свойств новой системы. Степень формальности программного проекта и его стадии будет влиять на необходимый уровень детализации, для каждого сценария использования. Диаграмма вариантов использования, описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования. Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки (рисунок 2.10) [7, 8, 9].

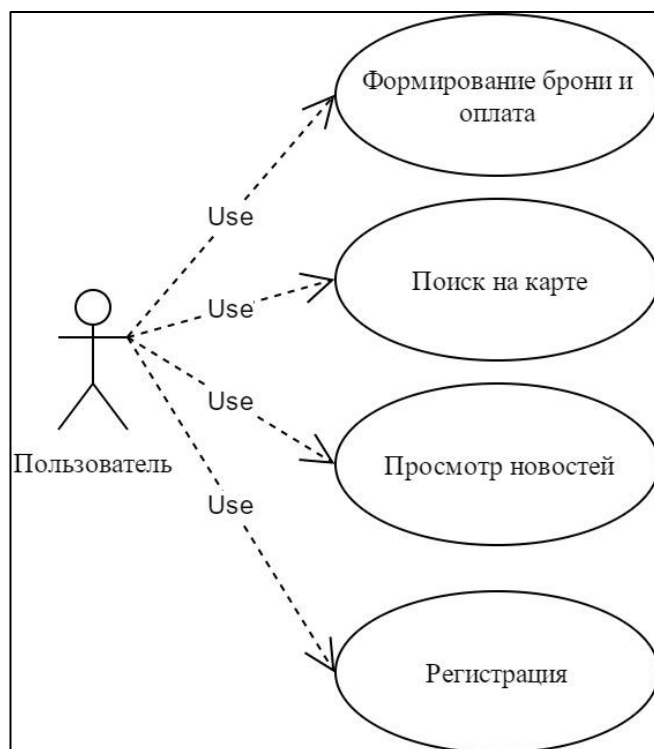


Рисунок 2.8 — Use-case диаграмма

## 2.2.1 UML-моделирование диаграммы деятельности

При моделировании поведения проектируемой системы часто возникает необходимость в детализации алгоритмической и логической реализации выполняемых системой операций. Для этого в языке UML используется диаграммы деятельности (рисунок 2.9) [7, 8, 9].

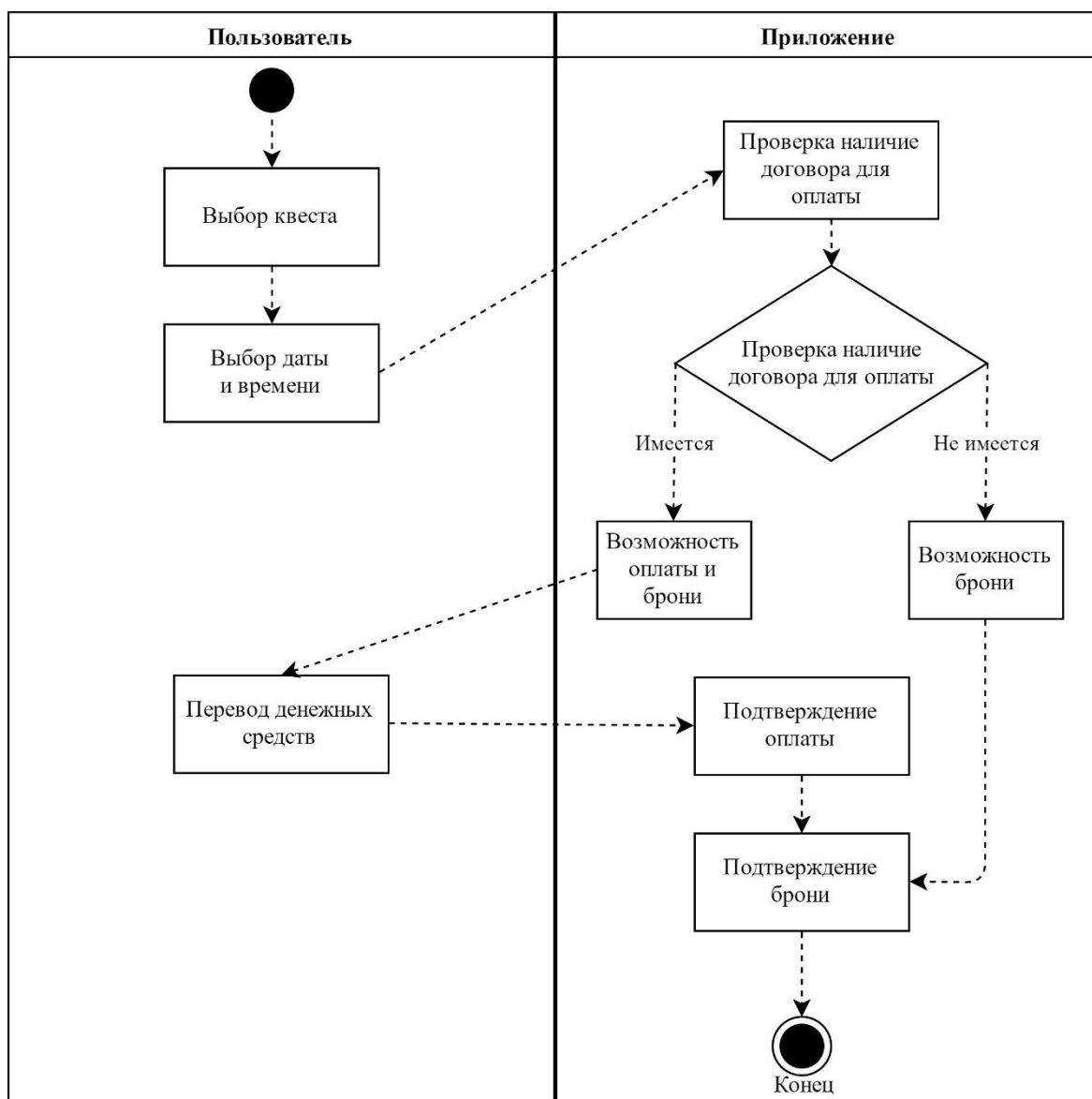


Рисунок 2.9 — Диаграмма деятельности. Формирование брони и оплаты

Поток событий:

- Пользователь выполняет выбирает в меню «Квесты».
- Пользователь выбирает заведение на карте.
- Приложение выводит подробную информацию о квесте.

- Пользователь нажимает кнопку «Хочу».
- Пользователь осуществляет выбор даты и времени.
- Приложение отправляет заявку на сервер для сохранения в базе данных.
- Приложение проверяет, есть ли договор на оплату заказа.
- Если имеется договор на оплату заказа с данным заведением, приложение выводит экран оплаты.
- Пользователь выбирает способ электронной оплаты (банковскую карту или электронный кошелек).
- Пользователь вводит данные счета для оплаты.
- Приложение выводит подтверждение, если перевод средств осуществлен.
- Приложение выводит подтверждение брони и оплаты.
- Альтернативный поток – договора на оплату заказа нету.
- Приложение выводит подтверждение бронирования.

### **2.2.2 UML-моделирование диаграммы классов**

С точки зрения языка программирования класс объектов можно рассматривать как тип данного, а отдельный объект - как переменную этого типа. Определение программистом собственных классов объектов для конкретного набора задач должно позволить описывать отдельные задачи в терминах самого класса задач (при соответствующем выборе имен типов и имен объектов, их параметров и выполняемых действий). Таким образом, объектно-ориентированный подход предполагает, что при разработке программы должны быть определены классы используемых в программе объектов и построены их описания, затем созданы экземпляры этих объектов и определено взаимодействие между ними (рисунок 2.10) [7, 8, 9].

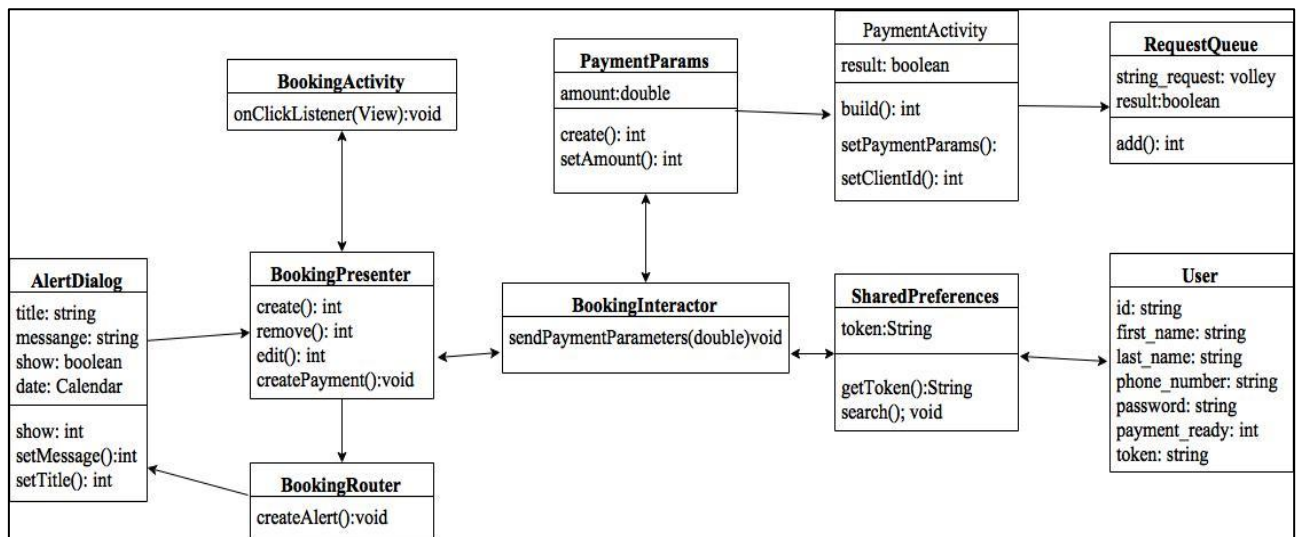


Рисунок 2.10 — Диаграмма классов

Классы:

- Класс User: содержит в себе информацию о клиенте, его логин и пароль.
- Класс AlertDialog: формируется во время процесса выбора заведения для подтверждения бронирования.
- Класс PaymentParams: формируется при подтверждении, желания пользователя оплатить заказ выводит стоимость заказа для оплаты.
- Класс PaymentActivity: формируется для вызова библиотеки Яндекс.Деньги и интерфейс оплаты.
- Класс RequestQueue: создаётся для отправки информации о заказе на сервер.
- Класс SharedPreference: репозиторий приложения.
- Класс BookingInteractor: содержит в себе бизнес-логику процесса бронирования/
- Класс BookingPresenter: определяет куда отдать управление.
- Класс BookingRouter: отвечает за переключение между различными экранами.
- Класс BookingActivity: содержит в себе окно выбора приложения.

### 2.2.3 UML-моделирование диаграммы последовательности

Диаграммы последовательности (sequence diagrams) предназначены для представления динамики поведения объектов, отображая передачу сообщений между соответствующими классами [7, 8, 9].

Прецедент: Бронирование и оплата (рисунок 2.11).

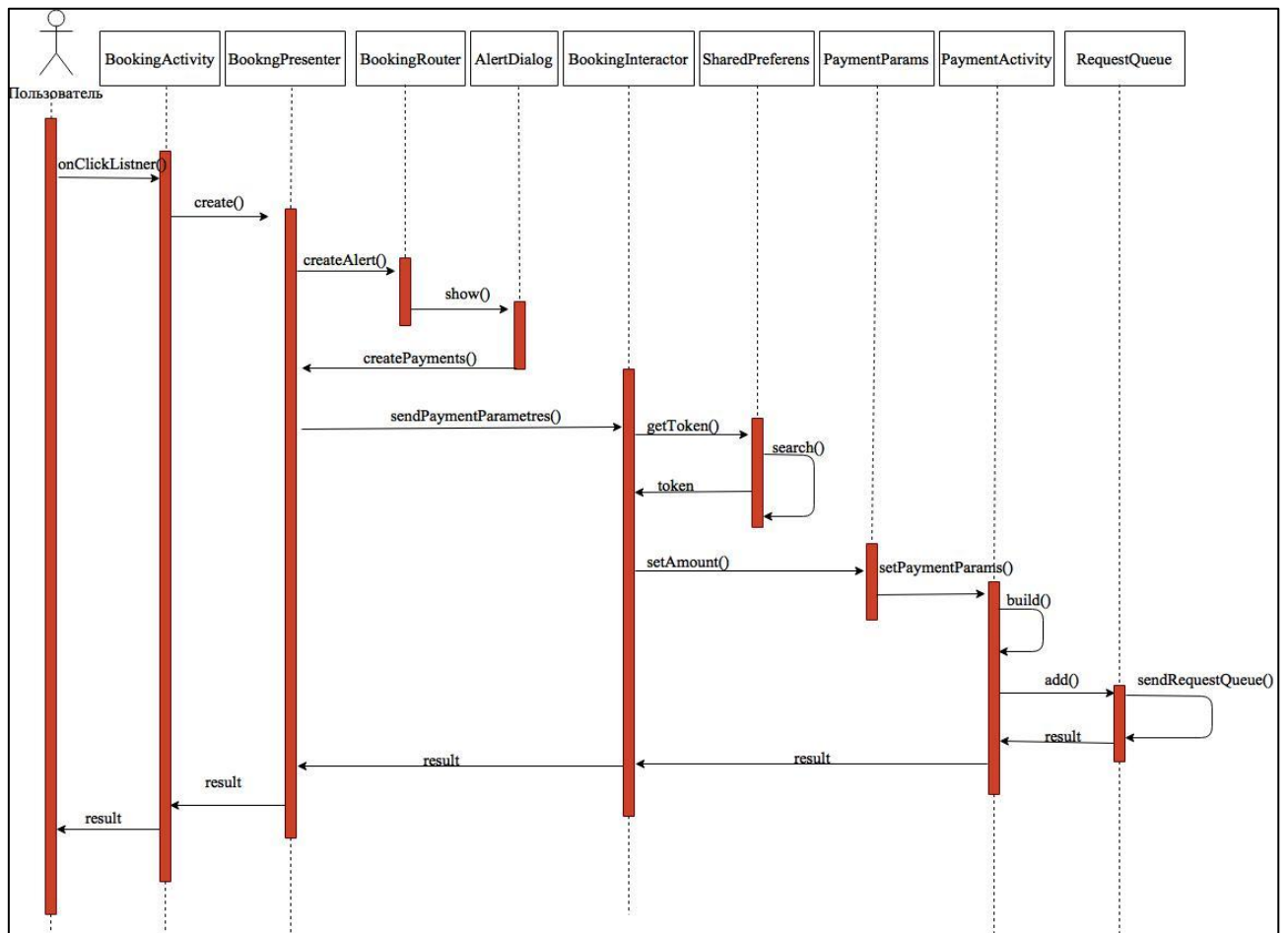


Рисунок 2.11 — Диаграмма последовательности

Краткое описание:

Действующее лицо этого прецедента пользователь.

Основываясь на диаграмме классов, выберем объекты, задействованные в данной диаграмме:

- User.
- AlertDialog.



- PaymentParams.
- PaymentActivity.
- RequestQueue.
- SharedPreference.
- BookingInteractor.
- BookingPresenter
- BookingRouter.
- BookingActivity.

Прецедент начинается, когда пользователь хочет забронировать место для участия в квесте:

- Объект «Пользователь», осуществляет регистрацию в приложение.
- «Пользователь» посылает объекту «User» сообщение «create», которое инициирует создание учетной записи пользователя.
- Объект «User» посылает «setMessage» и «setTitle» объекту «AlertDialog», что означающие выбор пользователя для бронирования заведения.
- Объект «AlertDialog» посылает «setAmount» объекту «PaymentParams», содержащий в себе стоимость заказа.
- Объект «PaymentParams» посылает «setPaymentParams» и «setClientid» объекту «PaymentActivity» для произведения оплаты.
- Объект «PaymentActivity» объекту «RequestQueue» переменную «result» сообщаящую о исполнении оплаты заказа.

#### **2.2.4 UML-моделирование диаграммы компонентов**

Любая программная система может считаться реализованной лишь в том случае, если она выполняет необходимый функционал. Для этого необходимо преобразовать программный код системы в исполняемые модули, библиотеки, файлы баз данных, и прочее. Таким образом, полный проект программной системы представляет собой совокупность логической и физической моделей.

В UML для физического представления моделей систем существует два вида диаграмм: диаграммы компонентов и диаграммы развертывания. Диаграмма компонентов – диаграмма, которая отображает компоненты системы и связи между ними (рисунок 2.12). Основной тип сущностей на диаграмме – это сами компоненты, а также интерфейсы, посредством которых указывается взаимосвязь между компонентами. На диаграмме компонентов применяются следующие отношения:

- реализации между компонентами и интерфейсами (компонент реализует интерфейс);
- зависимости между компонентами и интерфейсами (компонент использует интерфейс).

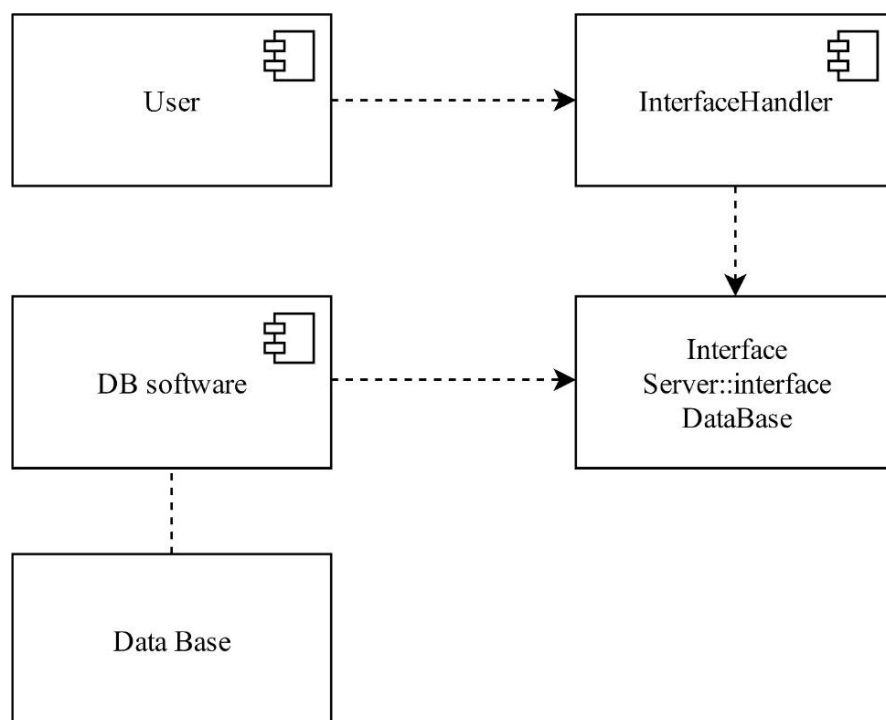


Рисунок 2.12 — Диаграмма компонентов

Компоненты:

- DBSoftware – компонент реализующий взаимодействие с БД системы;
- User – компонент для программного представления пользователя;

- InterfaceHandler – компонент реализующий графический интерфейс на стороне пользователя;
- IS Interface – интерфейс информационной системы.

### 2.2.5 UML-моделирование диаграммы развертывания

Диаграмма развертывания отображает способ взаимодействия компонентов с аппаратными средствами в физической системе, а также соединение аппаратных средств между собой (рисунок 2.13) [12, 13].

Клиентская часть:

- InterfaceHandler;
- User.

Серверная часть:

- DB Software;
- DataBase;
- IS Interface.

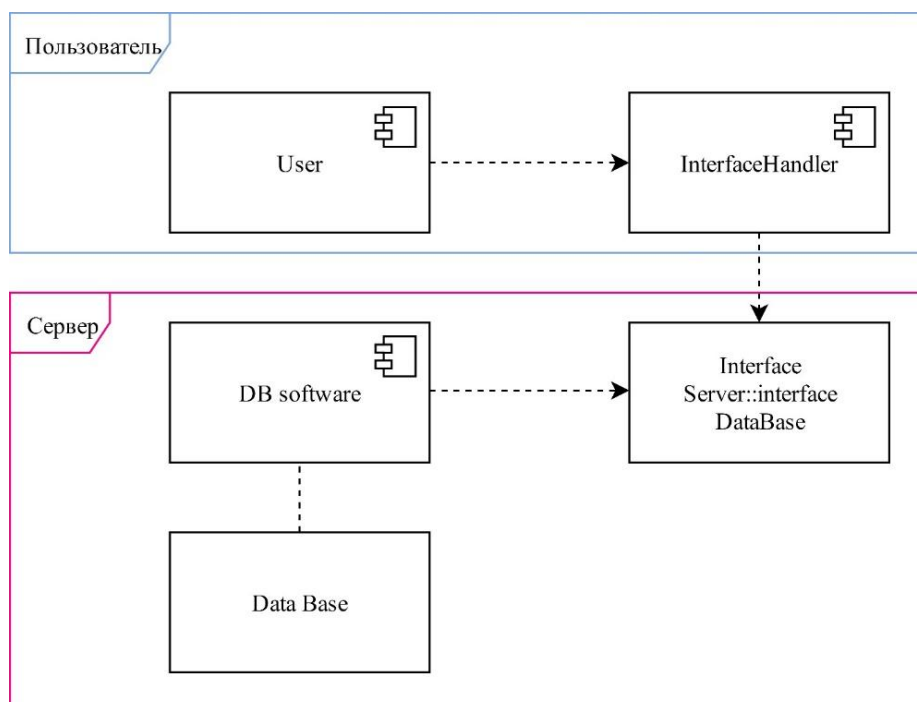


Рисунок 2.13 — Диаграмма развертывания

## **2.3 Вывод к главе 2**

Спроектирована UML-модель мобильного приложения CityQuest на базе операционной системы Android. Представлены Use-case диаграммы, деятельности, классов, последовательности, компонентов и развертывания. Изображены алгоритмы работы мобильного приложения.

## Глава 3 Разработка приложения

При разработке приложения используется язык Java на платформе Android Studio и библиотеки: «Яндекс.Деньги», «Google Maps API», «Material Design».

«Android Studio» — это интегрированная среда разработки (IDE) для работы с платформой Android. «Android Studio» основанная на программном обеспечении IntelliJ IDEA от компании JetBrains, официальное средство разработки Android приложений. Данная среда разработки доступна для Windows, OS X и Linux.

Библиотека «Яндекс.Деньги» — позволяет проводить операции с банковскими картами (Visa, MasterCard и Maestro), электронными кошельками (Яндекс.Деньги, Qiwi и WebMoney). «API Яндекс.Денег» для приложений позволяет использовать большую часть функций сервиса для разработки своих платёжных решений [9].

«Google Maps API» — интерфейс автоматически управляет доступом к серверам Google Карт, загрузкой данных, отображением карт и реакцией на жесты, выполняемые на картах [15].

«Material Design» представляет собой комплексную концепцию создания визуальных, движущихся и интерактивных элементов для различных платформ и устройств [13].

### 3.1 Интерфейс программы

Интерфейс программы должен быть простым и понятным для пользователя.

Реализованные следующие функциональные требования:

- функция регистрация в системе, с помощью номера телефона;
- функция обмена информации с сервером;
- функция авторизации в системе;
- функция просмотра карты квестов;

- функции просмотр подробной информации о квест комнате;
- функция бронирование квеста на определенную дату и число;
- функция оплаты.

### 3.1.1 Регистрация в приложении

Регистрации в приложении происходит путем ввода данных, после ввода приложение отправляет на сервер сообщение, при отправки данные кодируется форматом «application/x-www-form-urlencoded».

Входные данные:

- телефонный номер;
- пароль.

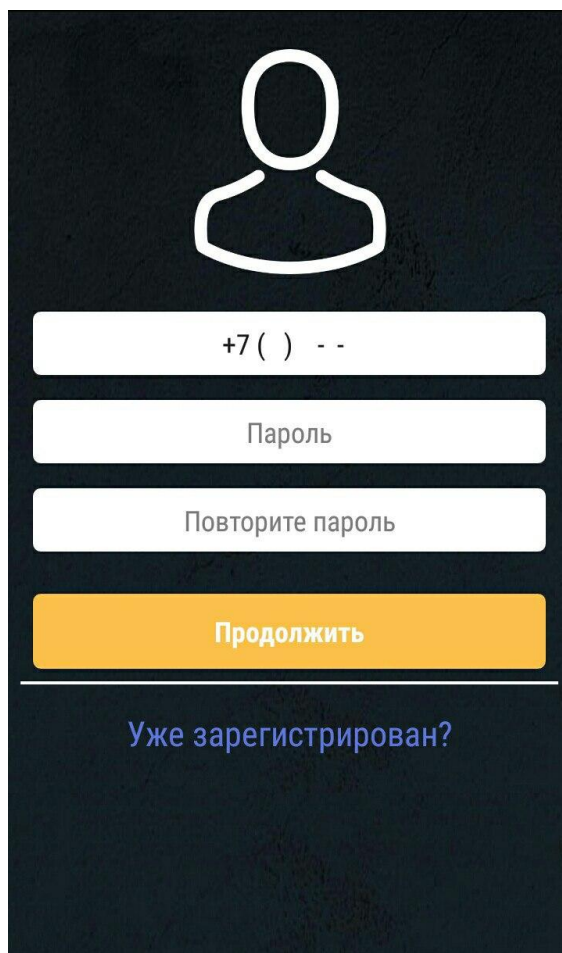


Рисунок 3.1 — Экран регистрация в приложение

После обработки сообщения сервер передает на указанный номер телефона уникальный код подтверждения для проверки подлинности вводимых данных (рисунок 3.2). Если номер телефона уже есть в базе данных, то приложение выдает ошибку.

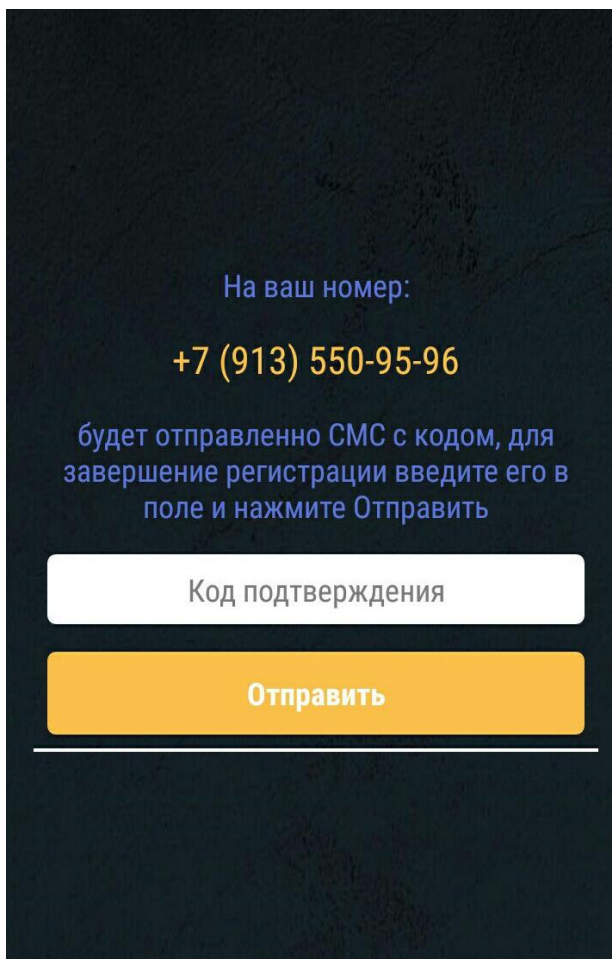


Рисунок 3.2 — Экран ввода кода подтверждения

### 3.1.2 Авторизация в приложении

Авторизация происходит путем ввода данных о своем аккаунте, которые отправляются для сверки их с базой данных пользователей.

Входные данные:

- телефонный номер;
- пароль.

После первоначального входа на мобильном устройстве сохраняется Token, который служит для автоматической повторной авторизации в приложение.

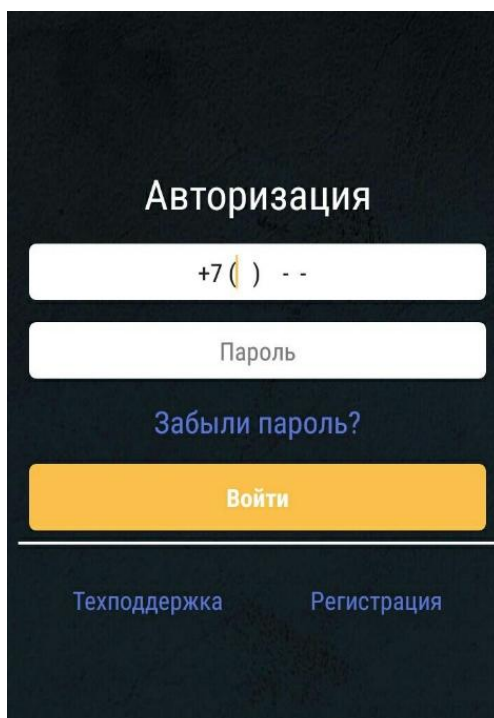


Рисунок 3.3 — Экран меню авторизации

### 3.1.3 Восстановление пароля

Если пользователь забыл пароль, существует восстановления пароля. На телефонный номер, внесенный в базу, отправляется сообщение с кодом подтверждения.



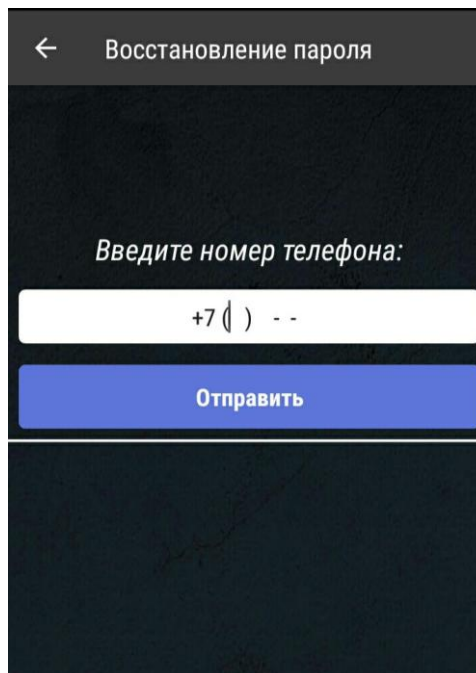


Рисунок 3.4 – Экран восстановление пароля

Если же данного телефонного номера нет в базе, выводится сообщение об ошибке. Далее на телефон приходит сообщение с кодом подтверждения для изменения пароля, таким же как при регистрации (рисунок 3.2).

### 3.1.4 Меню приложения

Меню приложение (рисунок 3.5) включает в себя следующие пункты:

- квесты (карта города Красноярска);
- новости и акции (Раздел со скидками);
- квест недели;
- мои брони;
- техподдержка;
- выйти.

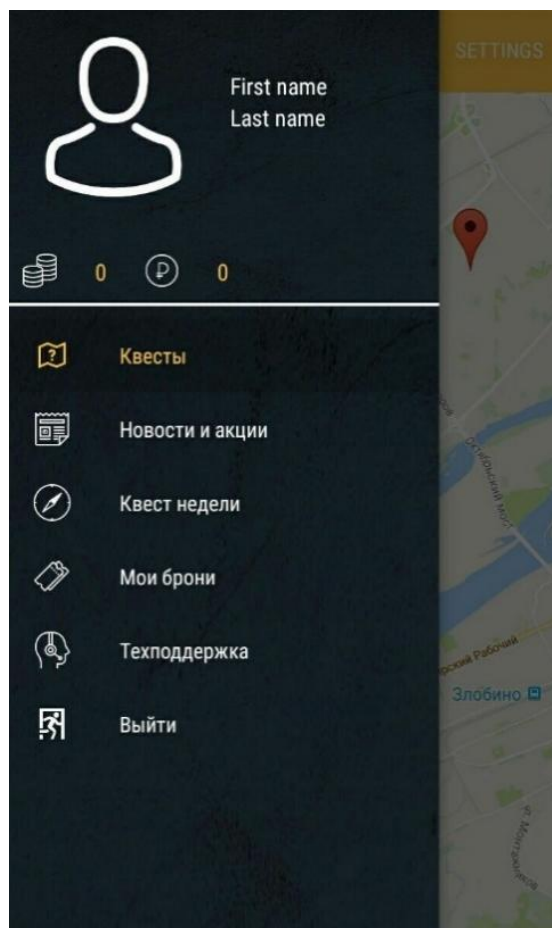


Рисунок 3.5 — Экран меню приложения

### 3.1.5 Карта квестов

Нажав на кнопку «Квесты» отображается карта Красноярска с отмеченными на ней маркерам, где находятся организации, которые проводят квесты (рисунок 3.6). Используется Google Maps API, путем превращения данных о карте в GeoJSON объект. GeoJSON - формат представления различных структур географических данных. GeoJSON поддерживает следующие геометрические типы: Point (точка), LineString (ломаная), Polygon (полигон), MultiPoint (мультиточка), MultiLineString (мультиломаная), MultiPolygon (мультиполигон) и GeometryCollection (коллекция геометрий) [10]. В разрабатываемом мобильном приложении, мы используем GEOJson объекты отражающие местоположение заведений на карте.

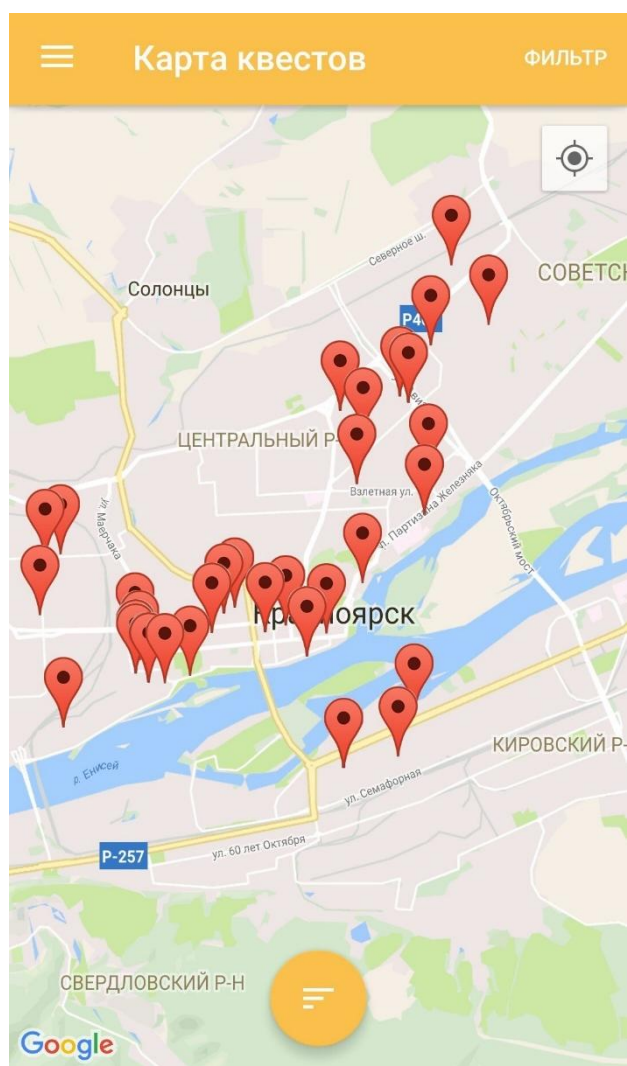


Рисунок 3.6 — Карта квестов



Рисунок 3.7 — Подробная информация о квестах

Отображения подробной информации о квестах происходит при нажатии на маркер, которое включает себя название, адрес, контактный телефон, краткое описание, длительность проведения и количество человек в группе для совместного прохождения (рисунок 3.7).

### 3.1.6 Бронирование

Процесс представляет собой основную функцию данного приложения. Происходит процесс на трех различных экранах. При выборе заведение приложение появляется Activity с полным описанием квеста, и кнопкой «Хочу» (осуществляет бронирование), изображенной на рисунке 3.8.



Рисунок 3.8 — Экран выбора квеста

Далее после нажатие кнопки появится интерфейс выбора даты и времени сеанса (рисунок 3.9).

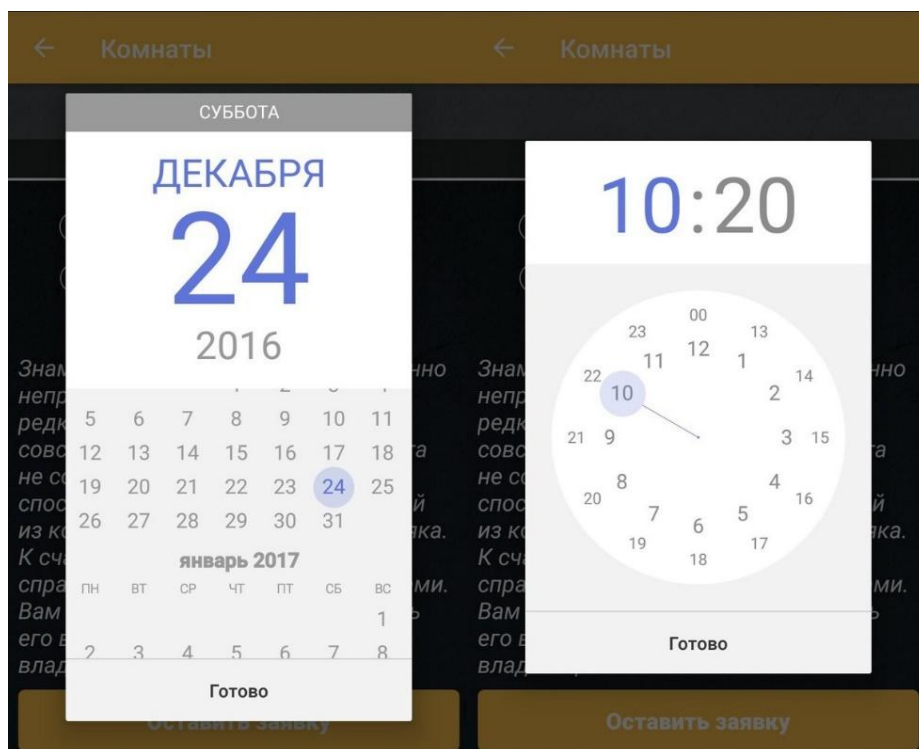


Рисунок 3.9 — Экран выбора даты и экран выбора времени

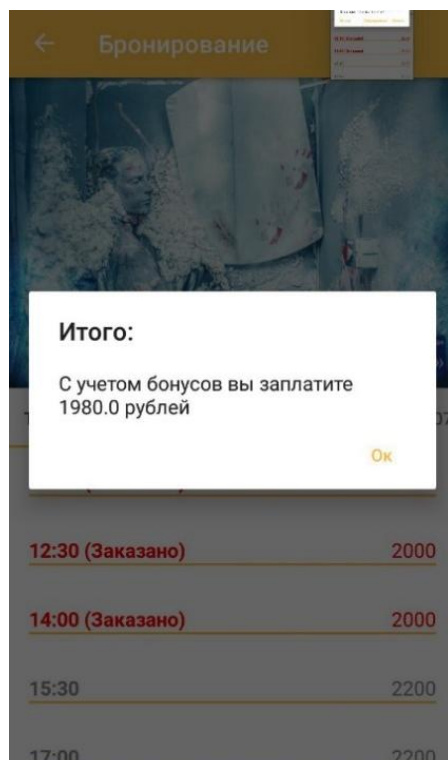


Рисунок 3.10 — Экран заказа

Функция оплаты происходит с помощью библиотеки Яндекс.Деньги. При оплате необходимо сообщить номер карты и сумму, далее осуществляется переход в следующее Activity для подтверждения операции путем ввода данных карты (рисунок 3.11).

Card Payment

Yandex Money

Перевод на счет [Redacted]

Recipient's account (transfer to a natural person)  
[Redacted]

To be paid  
1 980 rub.

Card number  
0000 1111 2222 3333

Valid until  
MM / YY

CVV code  
123

Pay

PCI DSS, MasterCard SecureCode, VERIFIED by VISA

Рисунок 3.11 — Экран оплаты

### 3.1.7 Мои брони

Функция отображение забронированных и оплаченных квестовых комнат. Отображает забронированные квесты пользователей и сумму оплаченных средств мероприятие (рисунок 3.12).

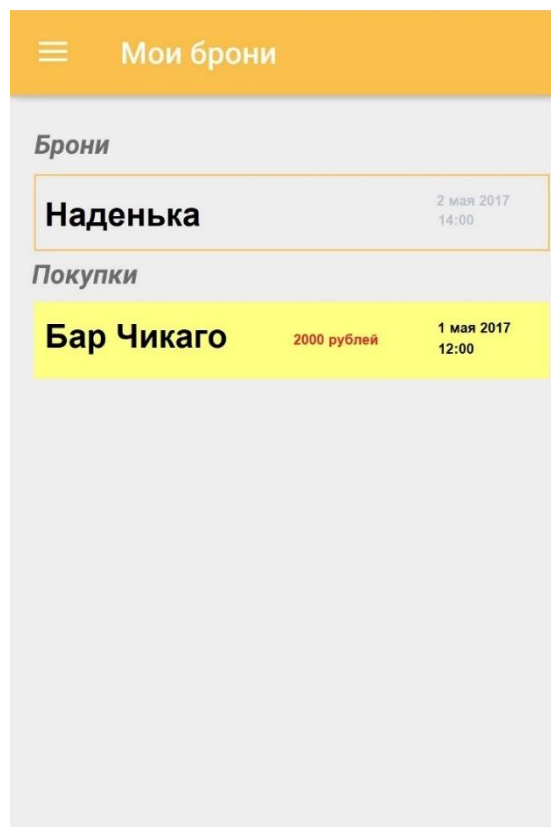


Рисунок 3.12 — Экран отображение заказов

### 3.1.8 Экран Техподдержки

Нажав на кнопку «Техподдержка» выводится интерфейс для отправки сообщений разработчику, где пользователь может сообщить о каких-либо неисправностях или возможных улучшения приложения (рисунок 3.13).

Экран содержит:

- окно ввода темы;
- окно ввода текста;
- интерфейс приложение к сообщению изображений.

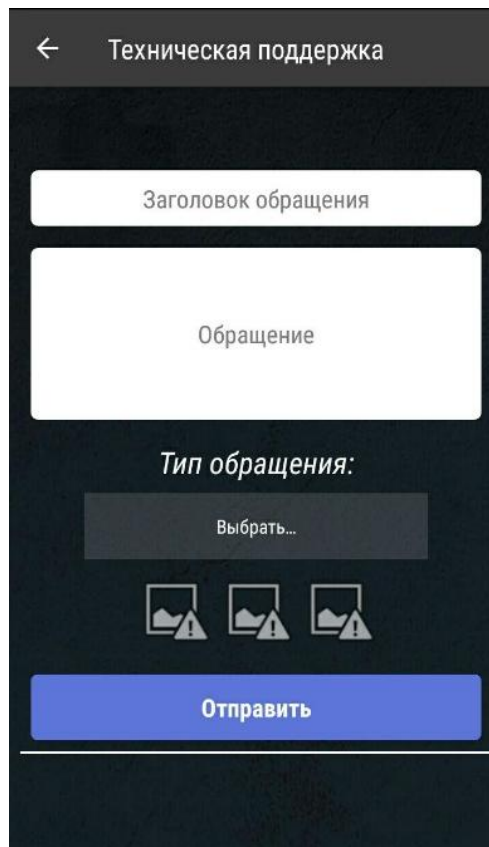


Рисунок 3.13 — Экран отправки сообщения в техподдержку

### **Вывод к главе 3**

Разработанная программа соответствует техническим требованиям, разработан интерфейс приложения.

В процессе разработки были удовлетворены все требуемые функции, выявленные в 1 главе, а именно:

- регистрация пользователей;
- отображение информации на карте;
- отображение подробной информации о квестах;
- бронирование;
- оплата.



## ЗАКЛЮЧЕНИЕ

В работе был проведен структурный анализ процесса составления формирования заявки, в результате чего были найдены пути автоматизации.

В результате обзора аналогичных мобильных приложений основной функционал приложения:

- Регистрация пользователей.
- Отображение информации на карте.
- Отображение подробной информации о квестах.
- Бронирование.
- Оплата.

При проектировании были созданы UML-модель мобильного приложение CityQuest на базе операционной системы Android. Представлены Use-case диаграммы, деятельности, классов, последовательности, компонентов и развертывания.

Приложение написано на языке Java, с использованием с использованием среды разработки Android Studio. Решены все поставленные задачи, разработан интерфейс приложения обеспечивающий выполнение всех функций, указанных в техническом задании. Реализованы требования к организации входных и выходных данных.

## **СПИСОК СОКРАЩЕНИЙ**

UML – Unified modeling language

CASE – Computer-aided software engineering

MVP – Model view presenter

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Методология функционального моделирования SADT. [Электронный ресурс] Режим доступа: [http://citforum.ru/database/case/glava2\\_2.html](http://citforum.ru/database/case/glava2_2.html)
2. Дейтел, П. Android для программистов: учебное пособие / Х. Дейтел, М. Морган – Санкт-Петербург: Питер, 2013. – 557 с.
3. Голощапов, А. Л. Google Android: книга / А. Л. Голощапов – Санкт-Петербург: БХВ-Петербург, 2013. – 268 с.
4. Мартин, Р. Чистый Код: учебное пособие / Р. Мартин – Санкт-Петербург: Питер, 2013. – 590 с.
5. ГОСТ 19.201-78 (ЕСПД) Техническое задание. Требования к содержанию и оформлению
6. Постановления правительства Красноярского края от 30 сентября 2013 г. № 504-П «Об утверждении государственной программы "Развитие информационного общества"»
7. Леоненков, А. В. «Самоучитель UML»: учебное пособие / А.В. Леоненков, – Санкт-Петербург: БХВ, 2004. – 275 с.
8. Шмуллер Д. Освой самостоятельно UML за 24 часа: учебное пособие / Д. Шмуллер. – Москва: Вильямс, 2005. – 150 с.
9. Ларман, К. Применение UML и шаблонов проектирования: учебное пособие / К. Ларман – Москва: Вильямс, 2004. - 624 с.
10. API для приложений. [Электронный ресурс] Режим доступа: <https://tech.yandex.ru/money/apps/?ncrnd=1291>
11. Спецификация формата GeoJSON. [Электронный ресурс] Режим доступа: [http://gis-lab.info/docs/geojson\\_ru.html](http://gis-lab.info/docs/geojson_ru.html)
12. Фаронов, В.А. Программирование на языке высокого уровня: учебное пособие /В.А. Фаронов. – Издательство: Питер, 2006. - 640 с.
13. Советов, Б. Я. Моделирование систем: учебное пособие / Б.Я. Советов – Москва: Вильямс, 2006. - 340 с.

14. Material Design для Android. [Электронный ресурс] Режим доступа: <https://developer.android.com/design/material/index.html?hl=ru>

15. Google Maps для каждой платформы [Электронный ресурс] Режим доступа: <https://developers.google.com/maps/?hl=ru>

16. Android Studio. [Электронный ресурс] Режим доступа: <https://developer.android.com/studio/index.html>

## **ПРИЛОЖЕНИЕ А**

### **Техническое задание**

ГОСТ 19.201-78. Техническое задание к содержанию и оформлению.

#### **1. Наименование системы**

Полное наименование системы: «мобильное приложение CityQuest».

Краткое наименование системы: CityQuest

#### **2. Наименование Заказчика и Исполнителя**

Заказчик: ИП Алексеев Владимир Петрович

Адрес: г. Красноярск ул. Дубровинского, 1 "А".

ИНН: 246511644328

ОГРНИП: 316246800075681.

Исполнитель: ФГАОУ ВО "СФУ".

Пользователи: люди, установившие приложение на смартфон с ОС «Android»

#### **3. Основания для разработки**

Технические требования, составленные в процессе опроса заказчика ИП Алексеев Владимир Петрович.

#### **4. Назначение разработки**

Программный продукт «мобильное приложение CityQuest» разрабатывается с целью, дать пользователям приложения возможность просмотра карточки заведения(квестов), бронирования и покупки мест в комнатах.

#### **5. Назначение и задачи системы**

Программный продукт «мобильное приложение CityQuest» предназначен для:

- Регистрация пользователей в системе.
- Просмотр квест комнат на карте и отображение подробной информации.
- Бронирование сеансов в заведениях (квестах).

## 6. Область применения

Программный продукт «мобильное приложение CityQuest» будет использоваться ИП Алексеев Владимир Петрович для продвижение своих услуг.

## 7. Требования к системе

«Мобильное приложение CityQuest» должен быть разработан в виде клиентской программы для мобильного устройства на базе операционной системы Android, отправляющая информацию на сервер для заполнения базы данных.

Система должна обеспечивать возможность выполнение следующих функций:

- Функция регистрация в системе, с помощью номера телефона.
- Функция обмена информации с сервером с помощью API.
- Функция авторизации в системе.
- Функция просмотра карты квестов;
- Функции просмотр подробной информации о квест комнате.
- Функция бронирование квеста на определенную дату и число.

Требования к организации входных и выходных данных.

Входные данные - внесение информации в БД:

- Телефонный номер.
- Пароль.
- Имя.
- Фамилия.
- Фотография.

## 8. Условия эксплуатации

Эксплуатация программы происходит в условия соответствующих с установленными в инструкции к мобильному устройству. На устройстве

должна быть установлена операционная система Android версии не ниже 4.3 (Jelly Bean).

## ПРИЛОЖЕНИЕ Б

### Графический материал

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт космических и информационных технологий  
Систем искусственного интеллекта

Проектирование и разработка мобильного приложения CityQuest на  
базе операционной системы Android

Руководитель:

Старший преподаватель кафедры СИИ К. В. Раевич

Выпускник:

А. А. Мазун

Рисунок Б.1 — слайд презентации №1

### Цель и задачи

Цель: Проектирование и разработка мобильного приложения CityQuest на базе операционной системы Android.

Задачи:

- 1) Структурный анализ информационных процессов, обзор существующих мобильных приложений.
- 2) Проектирование мобильного приложения CityQuest.
- 3) Разработка мобильного приложения CityQuest.

Рисунок Б.2 — слайд презентации №2



# Актуальность

- 1) Постановления правительства Красноярского края от 30 сентября 2013 г. N 504-П об утверждении государственной программы «Развитие информационного общества».
- 2) Разработка ведется по договор №152/л от 10 февраля 2017.
- 3) Автоматизация приема заявок.

2

Рисунок Б.3 — слайд презентации №3

## Методология структурного анализа и проектирования (SADT-модели)



Рисунок 1 — Модель «AS-IS»

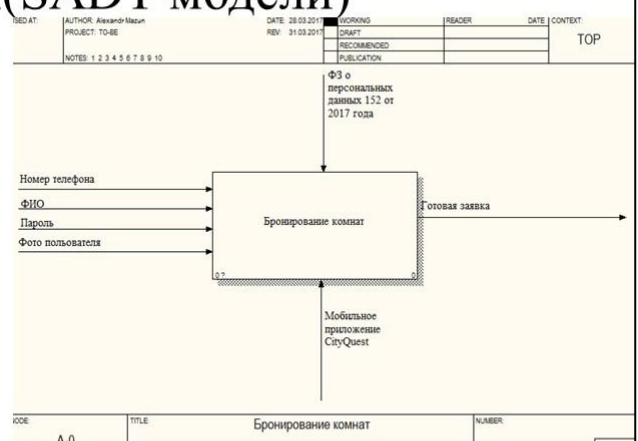


Рисунок 2 — Модель «TO-BE»

3

Рисунок Б.4 — слайд презентации №4

## Диаграммы декомпозиции

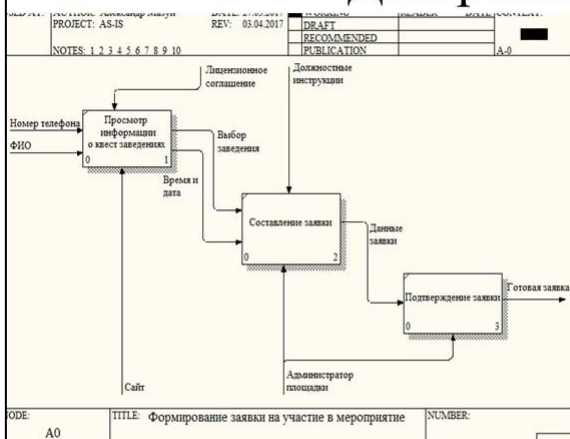


Рисунок 3 — Декомпозиция блока «Формирование заявки на участие в квесте» (AS-IS)

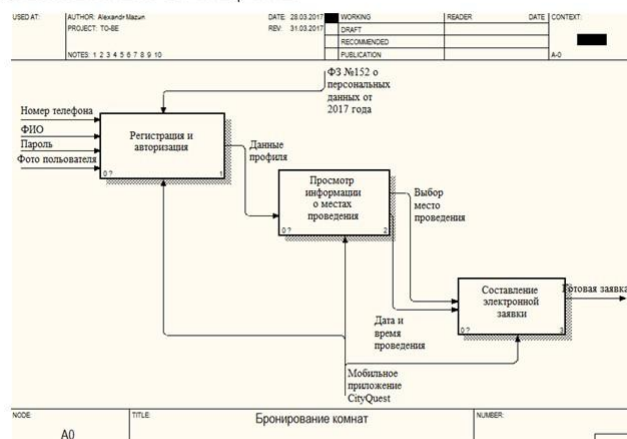


Рисунок 4 — Декомпозиция блока «Формирование заявки на участие в квесте» (TO-BE)

4

Рисунок Б.5 — слайд презентации №5

## Обзор приложения «СибКвест»

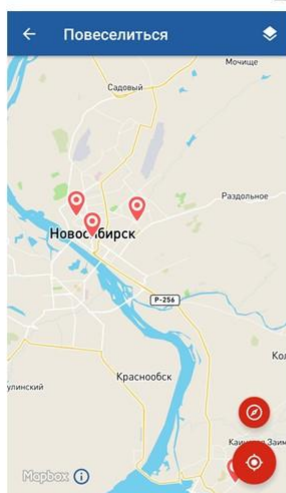


Рисунок 5 — Окна просмотра квестов на карте



Рисунок 6 — Окна просмотра квестов в виде списка

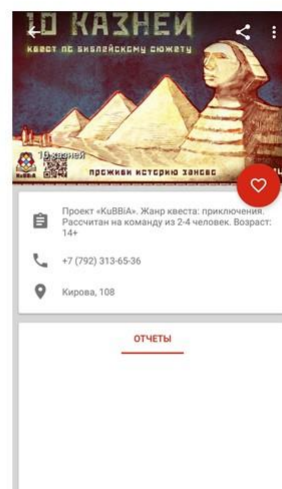


Рисунок 7 — Окно просмотра подробной информации

5

Рисунок Б.6 — слайд презентации №6

## Обзор приложения «Ловушка»



Рисунок 8 — Меню приложения



Рисунок 9 — Экран отображения информации

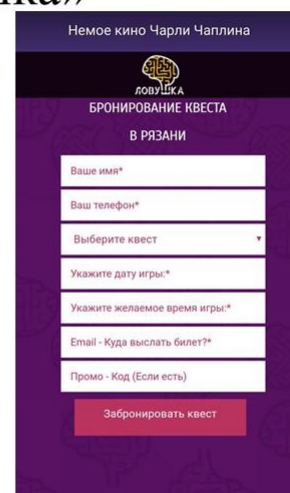


Рисунок 10 — Экран бронирования

6

Рисунок Б.7 — слайд презентации №7

## Обзор рассмотренных систем

Таблица 1 – Сравнение функционала

Приложение	Функции					
	Авторизация	Просмотра на карте	Просмотр подробной информации	Отслеживание геопозиции пользователя	Бронирование	Оплата
«СибКвест»	Не имеет	есть	есть	есть	отсутствует	отсутствует
«Ловушка»	Не имеет	отсутствует	есть	отсутствует	есть	отсутствует
«CityQuest»	С помощью номер мобильного телефона и пароля	есть	есть	есть	есть	есть

7

Рисунок Б.8 — слайд презентации №8

## Требования к мобильному приложению

- Функция регистрация в системе, с помощью номера телефона.
- Функция авторизации в системе.
- Функция просмотра карты квестов.
- Функции просмотр подробной информации о квест комнате.
- Функция бронирование квеста на определенную дату и число.
- Функция оплаты бронирования.

8

Рисунок Б.9 — слайд презентации №9

## Архитектура VIPER

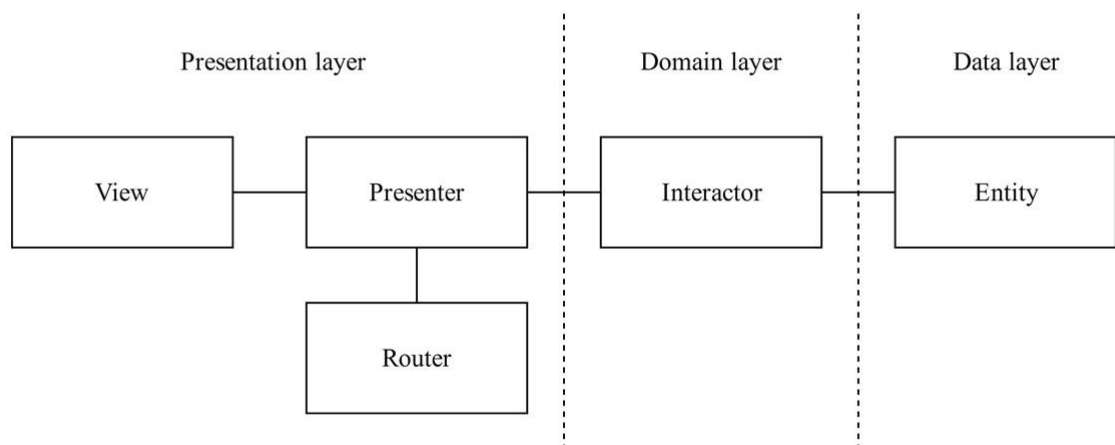


Рисунок 11 — Архитектура VIPER

9

Рисунок Б.10 — слайд презентации №10

# Uml - диаграммы

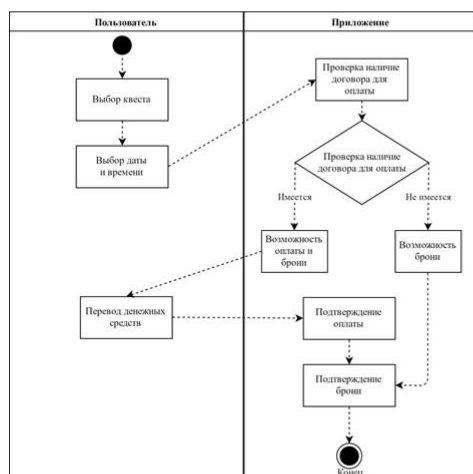


Рисунок 12– Диаграмма деятельности

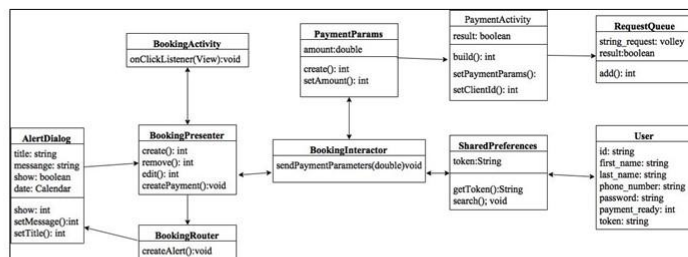


Рисунок 13 – Диаграмма классов

10

Рисунок Б.11 — слайд презентации №11

# Uml - диаграммы

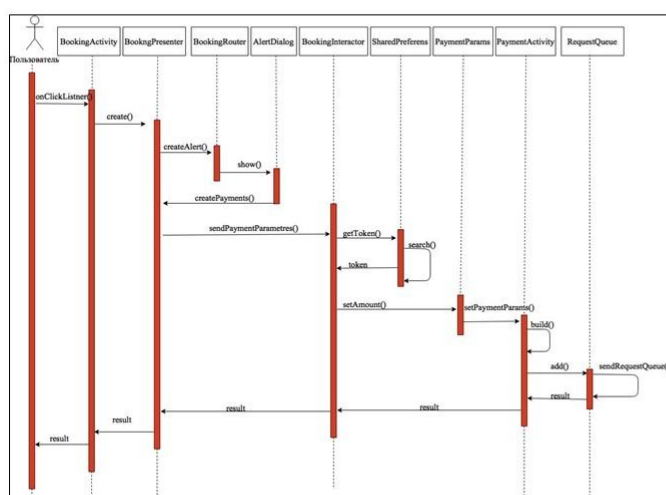


Рисунок 14 – диаграмма последовательности

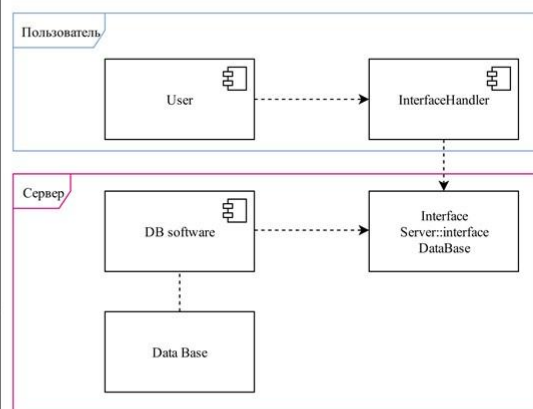


Рисунок 15 – диаграмма развертывания

11

Рисунок Б.12 — слайд презентации №12

## Разработка программы

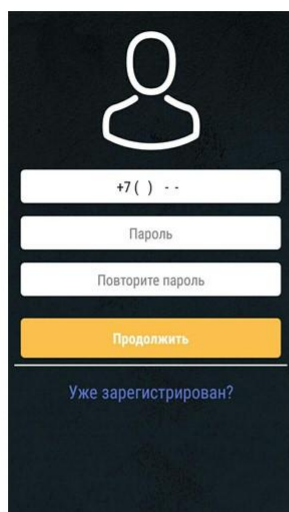


Рисунок 16 – Экран регистрации в приложение

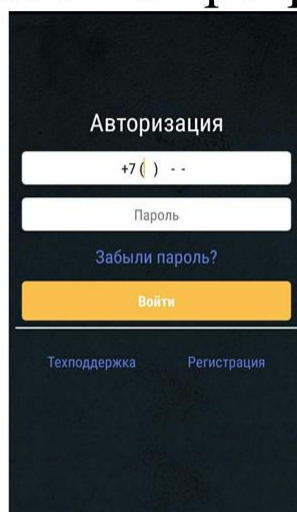


Рисунок 17 – Экран меню авторизации

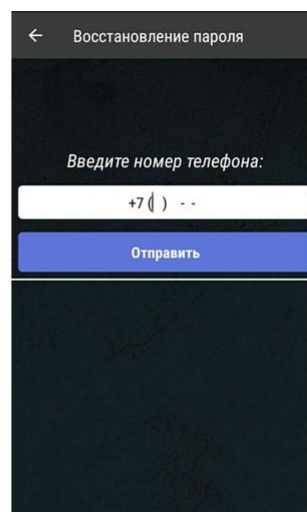


Рисунок 18 – Экран восстановления пароля

12

Рисунок Б.13 — слайд презентации №13

## Разработка программы

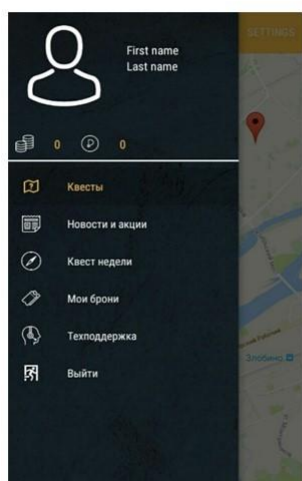


Рисунок 19 – Экран меню приложения

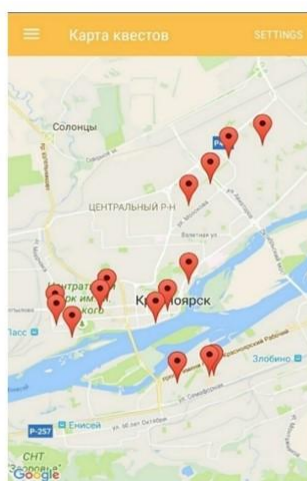


Рисунок 20 – Карта квестов

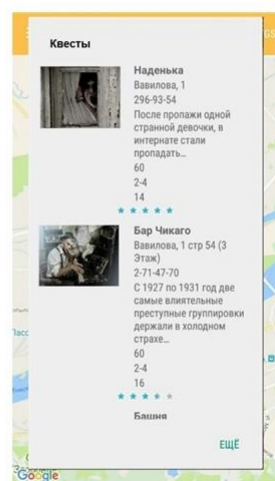


Рисунок 21 – Подробная информация о квестах

13

Рисунок Б.14 — слайд презентации №14



## Разработка программы



Рисунок 22 – Экран выбора квеста

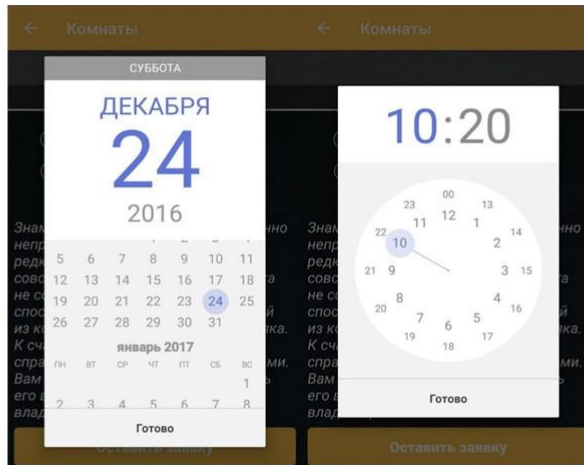


Рисунок 23 – Экран выбора даты и экран выбора времени



Рисунок 24– Экран заказа

14

Рисунок Б.15 — слайд презентации №15

## Разработка программы

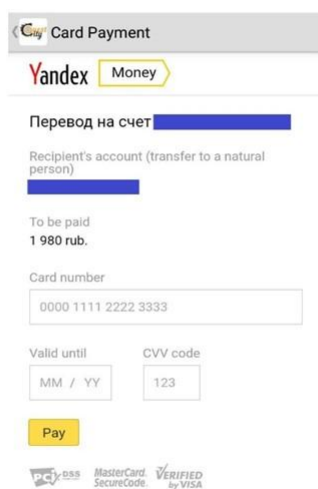


Рисунок 25 – Экран оплаты

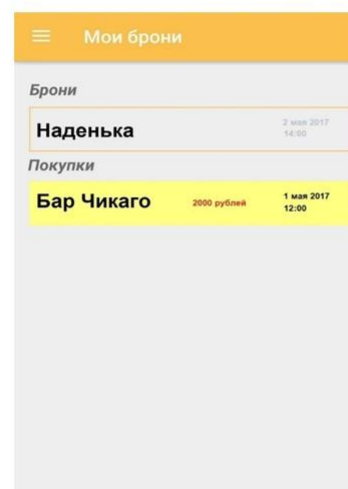


Рисунок 26– Экран отображение заказов

15

Рисунок Б.16 — слайд презентации №16

## Заключение

В ходе выпускной квалификационной работы по направлению 09.03.02 «Информационные системы и технологии» был проведен структурный анализ информационных процессов (формирование заявки) и обзор существующих мобильных приложений.

Этапы проектирования мобильного приложения CityQuest отображены с помощью UML-диаграмм.

Приложение реализовано на языке Java на платформе Android Studio. Решены поставленные задачи, разработан интерфейс, приложение обеспечивает выполнение всех функций, указанных в техническом задании.

Данное приложение размещено на платформе Play Market.

16

Рисунок Б.17 — слайд презентации №17